

BİL 372 PROJE RAPORU

Deniz Öner, Şayegan Oya Erdayı, Şükran Saygılı, Yavuz Algün

1.Rapor Tanıtımı

Bu raporda oluşturulan insan kaynakları yönetim sisteminin içeriğinden, çalışma şeklinden bahsedilmiştir. Oluşturulan bu insan kaynakları sistemine ait ekran görüntüleri ve ilgili ekranın nasıl çalıştığı, ekranda kullanılan SQL komutlarından bahsedilmiştir. Projenin içeriğini daha iyi tanıtabilmek için hipo diagramı, data flow diagramı verilmiştir.

2.Proje Tanıtımı

Oluşturulan bu projede bir İnsan Kaynakları Yönetim Sistemi tasarlanmıştır. Projede kullanılan dil C# MVC olup, Database olarak Azure Database kullanılmıştır. Sisteme admin veya personel olarak giriş yapılabilir. Admin tarafından giriş yapan kullanıcıların ekranları, yapabilecekleri ile personel tarafından giriş yapan kullanıcıların yaptıkları farklılık gösterir.

Personel

Oluşturulan bu sistem kurumsal firmalarda personellerin ve insan kaynaklarının kullanabileceği bir sistemdir. Personel sistemi açar. Kendi kullanıcı adı ve şifresiyle login olma işlemini gerçekleştirir. Personel sistemden kendi kişisel bilgilerini “Personel information” sekmesinden görüntüleyebilir. Adını, soyadını, email adresini, yaptığı işi, bağlı olduğu departmanı, maaş bilgisini, kendine ait olan sertifikaları ve çalıştığı projeleri görüntüleyebilir. Personel kendi maaş bilgisinin bilgilerini görüntülemesinin yanı sıra eski maaş bilgisini maaş planım sekmesinden görüntüleyebilir. Personel sistemde kendine ait sertifika bilgisini görüntüleyebilir. Personel maaş bilgisini değiştiremez. Personel kendi için izin alır.

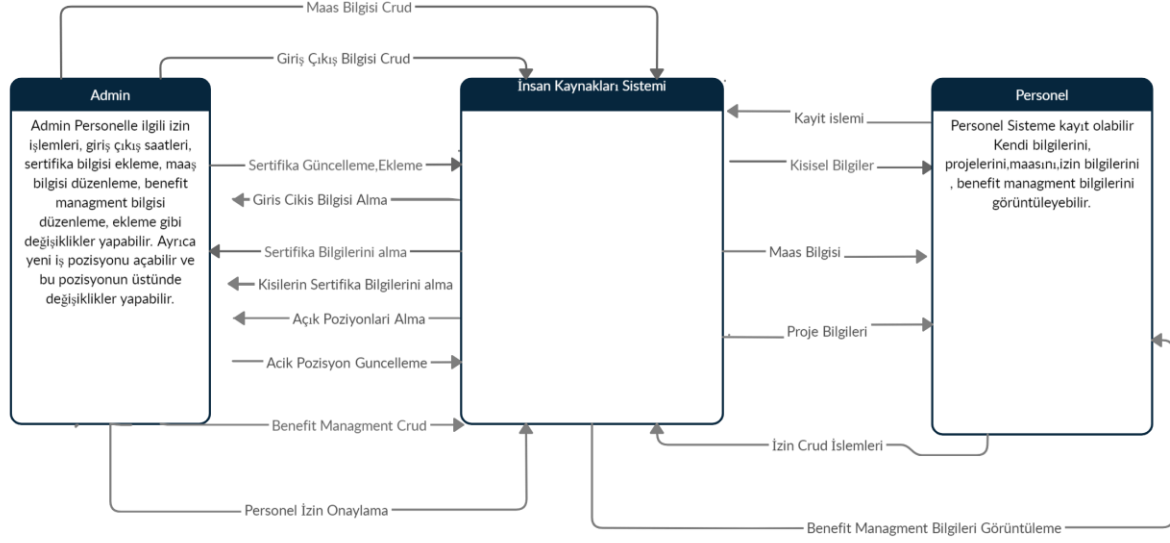
Admin

Bir insan kaynakları uzmanı sisteme kendi bilgilerini girerek giriş yapar. Sistemden yeni iş pozisyonu oluşturabilir. Buradan oluşturulan yeni iş aç sekmesinin altında iş başlığı, iş tanımı, istenilen yetenekler gibi verileri doldurabilir. Böylece açık bir iş pozisyonu yaratır. Mevcut açık pozisyonlar listesini görüntüleyebilir ve açık pozisyonu kapatabilir.

Time&attendance ekranından görüntülemek istediği personele ait giriş çıkış saatleri bilgilerini görüntüleyebilir ve bunları silebilir. Benefit Management ekranından görüntülemek istediği personele ait sağlık sigortası, sağlık bilgisi bilgilerini görüntüleyebilir.

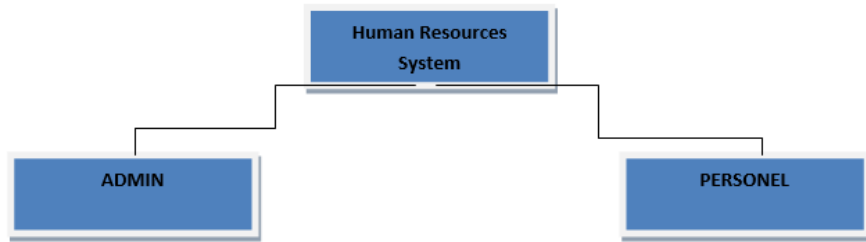
Learning&Professional Development ekranından sertifika ekleyip bunu istediği personele atayabilir. Bu yaptığı değişiklikleri görüntüleyip, istediği bilgileri silebilir. Admin sistemde yer alan projeleri ve bağlı olduğu departmanları görüntüleyebilir. Yeni proje ekleyebilir. Eklediği bir projeyi departmana bağlar. Projeleri sistemde yer alan personellere bağlayabilir. Admin sisteme yeni bir departman ekleyebilir ve bu departman üzerinde değişiklik yapabilir.

3.Projenin Data Flow Diyagramı

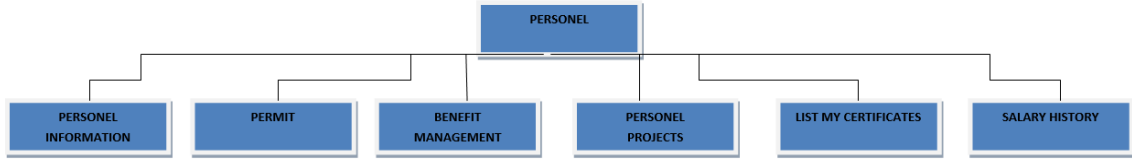


4.Projenin HIPO Diagramı

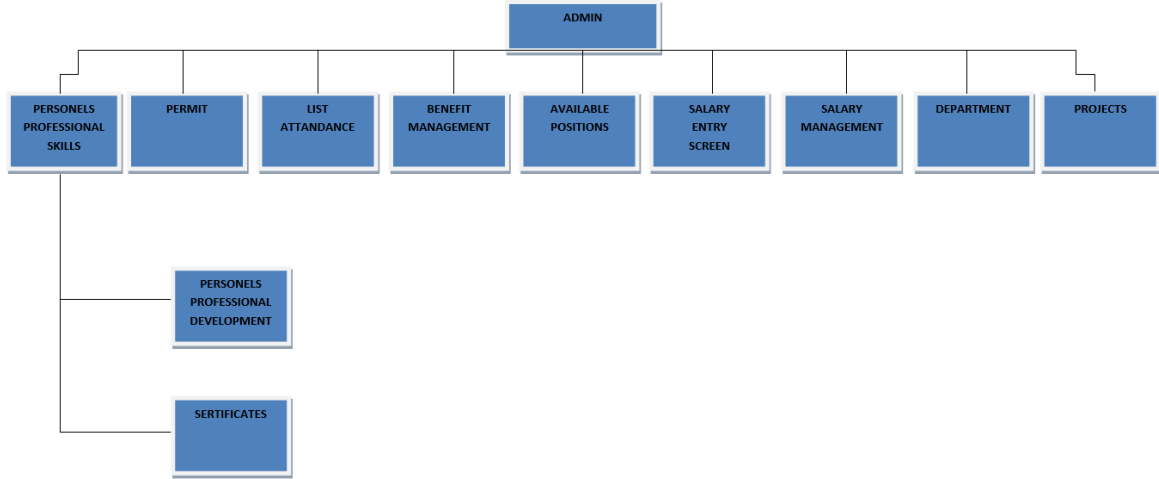
HIPO DIAGRAM



Personel



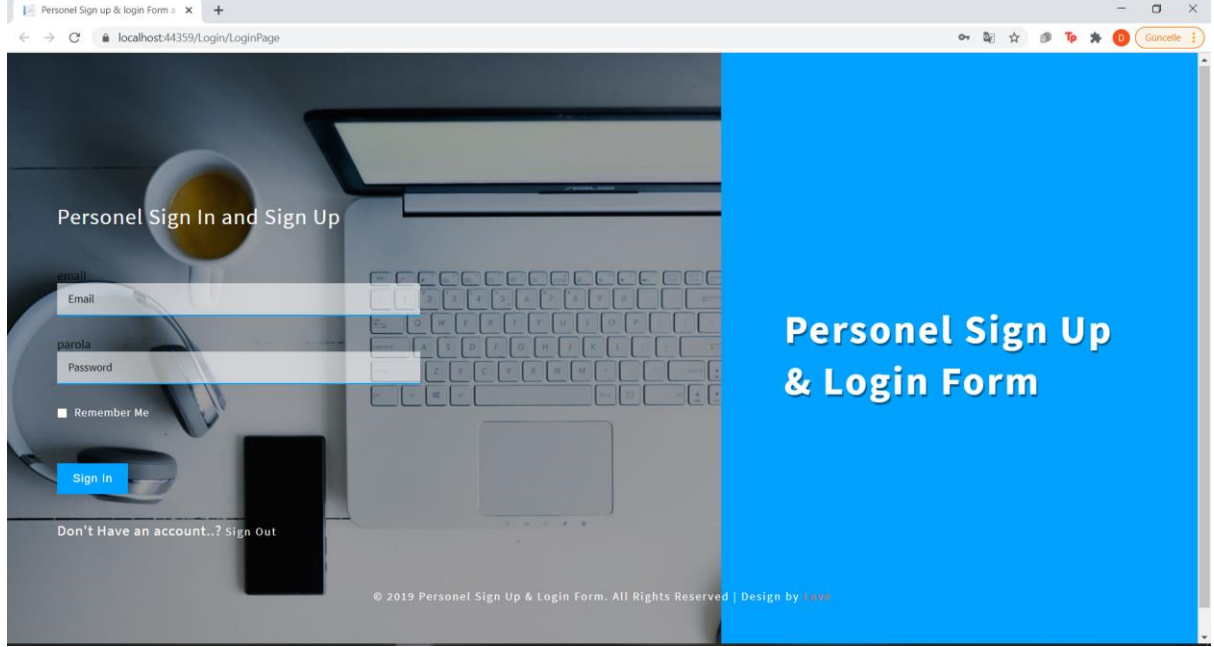
Admin



5.Projede Yer Alan Ekranların Tanıtımı ve Kullanılan Sql Komutları

Personel Girişi

1.Login Ekranı



Login Sayfasının Ekran Görüntüsü

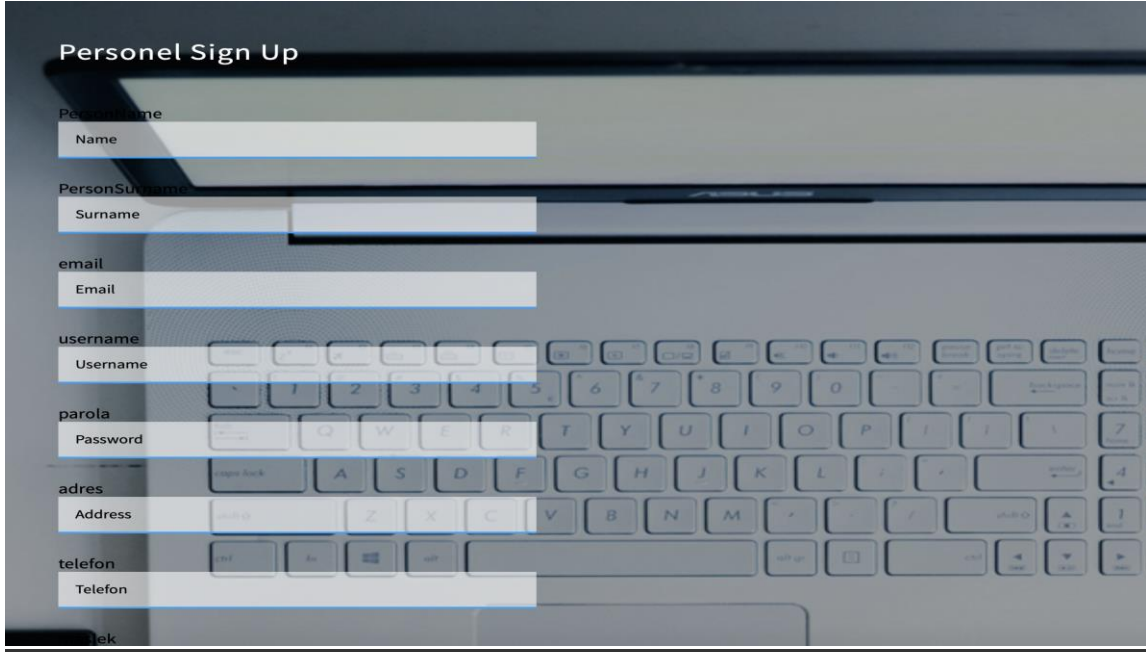
Kullanıcı sisteme giriş yapınca login ekranını görüntüler. Buradan personel kendi bilgilerini girerek sisteme giriş yapabilir, aynı şekilde admin burdan kendi bilgilerini girerek admin tarafına giriş yapabilir.

Sisteme kullanıcı bilgilerini girince sql komutu ile böyle bir kullanıcı olup olmadığı kontrolü sağlanır ve eğer yoksa tekrar giriş yapılması istenir. Eğer böyle bir kullanıcı varsa, giren kişi admin ise admin ekranına, personel ise personel ekranına giriş yapar.

SQL KOMUTU

```
var person = DbContext.Personel.Where(x => x.email == personelModel.email && x.parola == personelModel.parola).FirstOrDefault();
```

2.Register



Personel Sign Up

PersonName
Name

PersonSurname
Surname

email
Email

username
Username

parola
Password

adres
Address

telefon
Telefon

Register Sayfasının Ekran Görüntüsü

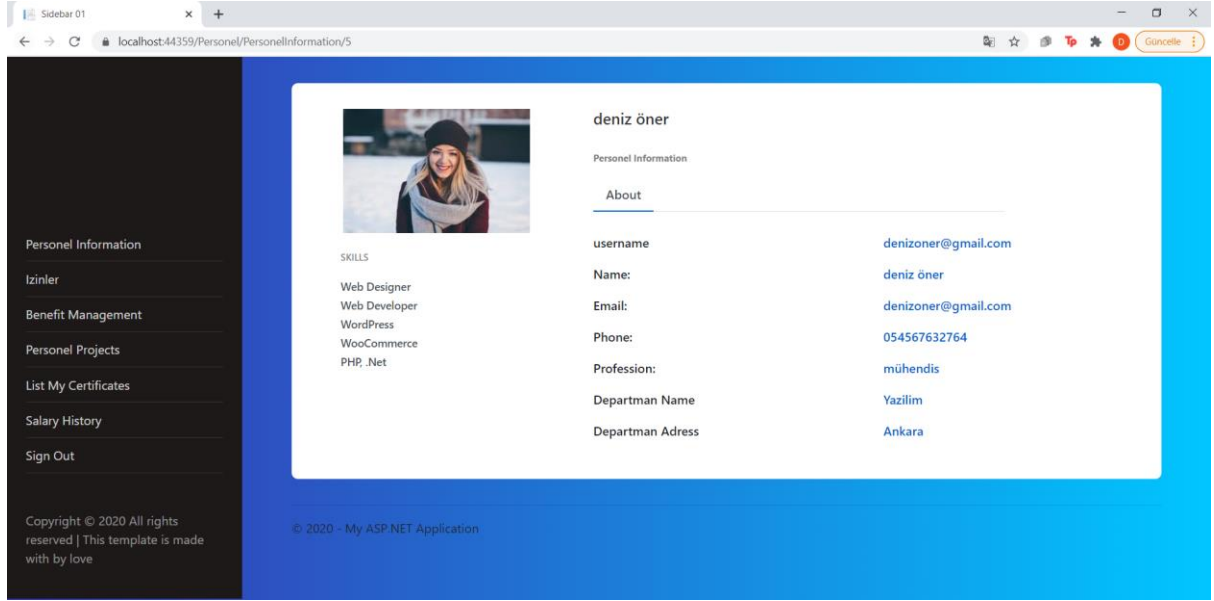
Kullanıcının eğer sisteme üyeliği yok veya öncesinde herhangi bir giriş accountu yoksa login sayfasındaki sign up alanına basarak bu ekrana giriş yapılabilir. Giriş yapılan bu ekranda kullanıcıya ait giriş bilgileri alınarak arka tarafta bir user account oluşturulur.

Burada kullanıcının sayfadaki forma girdiği bilgileri sql komutu ile database'de tuttuğumuz personel tablomuza kaydedilir. Böylece kullanıcı için sistemde bir account oluşturulmuş olur.

SQL KOMUTU

```
DbContext.Personel.Add(personelModel);  
DbContext.SaveChanges();
```

3. Personel Bilgileri Görüntüleme Ekranı



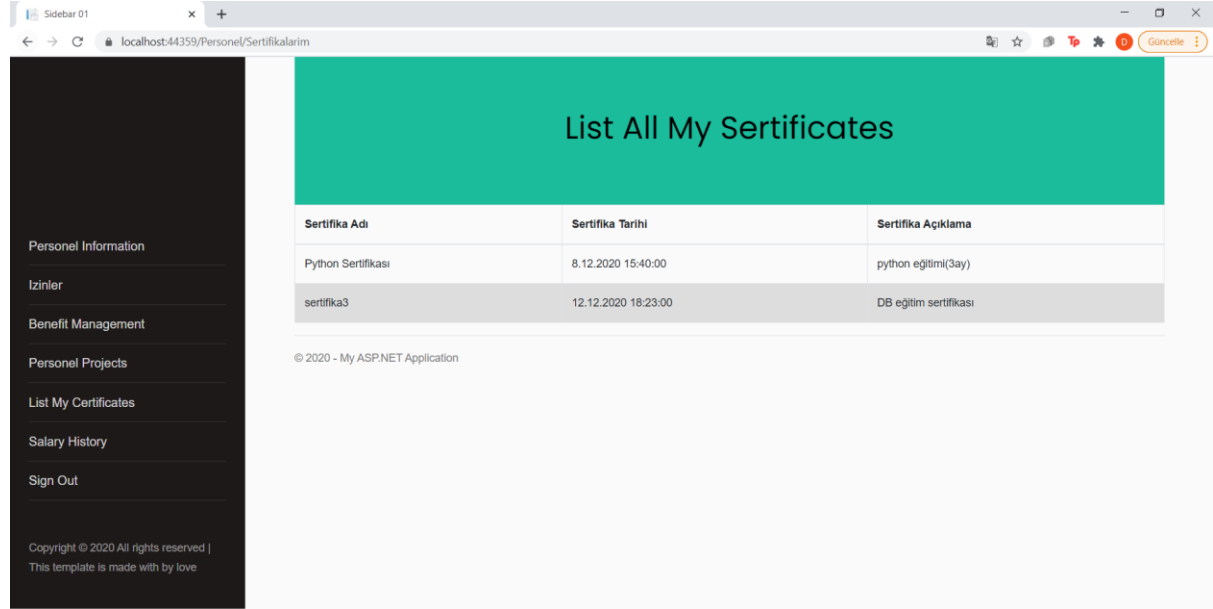
Personel Information Ekran Görüntüsü

Kullanıcı sisteme giriş yapınca kendi bilgilerinin yer aldığı ekrana giriş yapar. Buradan kendi kişisel bilgilerini görüntüleyebilir. Ekranın sol tarafında yer alan butonlardan kendiyle ilgili diğer sayfalara geçiş yapılabilir.

SQL KOMUTU

```
var person = DbContext.Personel.Where(x => x.ID == id).FirstOrDefault();  
var departman = DbContext.Departmes.Where(x => x.DepartmentID ==  
person.dno).FirstOrDefault();
```

4. Personelin Kendi Sertifikalarını Görüntüleme Ekranı



List All My Certificates Ekranı

Kullanıcı sisteme giriş yapınca “List All My Certificates” sekmesinden kendi sertifikalarını görebileceği ekrana giriş yapar. Buradan admin tarafından kendisine atanmış sertifikaları görüntüleyebilir. Ekranın sol tarafında yer alan butonlardan kendiyle ilgili diğer sayfalara geçiş yapılabilir.

SQL Komutu

```
List<Learn_DevelopmentModel> learn_Developments =  
DbContext.Learn_DevelopmentModels.Where(x => x.person_id == id).ToList();  
List<SertifikaModel> sertificate = DbContext.SertifikaModels.ToList();  
List<SertifikaModel> newsertificate = new List<SertifikaModel>();  
var viewModel = new SertifikaViewModel();
```

```
foreach (var value in learn_Developments)  
{  
viewModel.SertifikaModels.Add((sertificate.Where(x => x.ID ==  
value.sertifika_id).FirstOrDefault()));  
}
```

5.Personelin Maaş Geçmişini Görüntüleme Ekranı

Salary History

#	Miktar	Tarih
1	30000	1.01.2021 00:00:00
2	28000	2.01.2021 00:00:00
3	1200	12.12.2020 00:00:00
4	20	12.12.2020 00:00:00
5	30000	15.12.2020 00:00:00

© 2020 - My ASP.NET Application

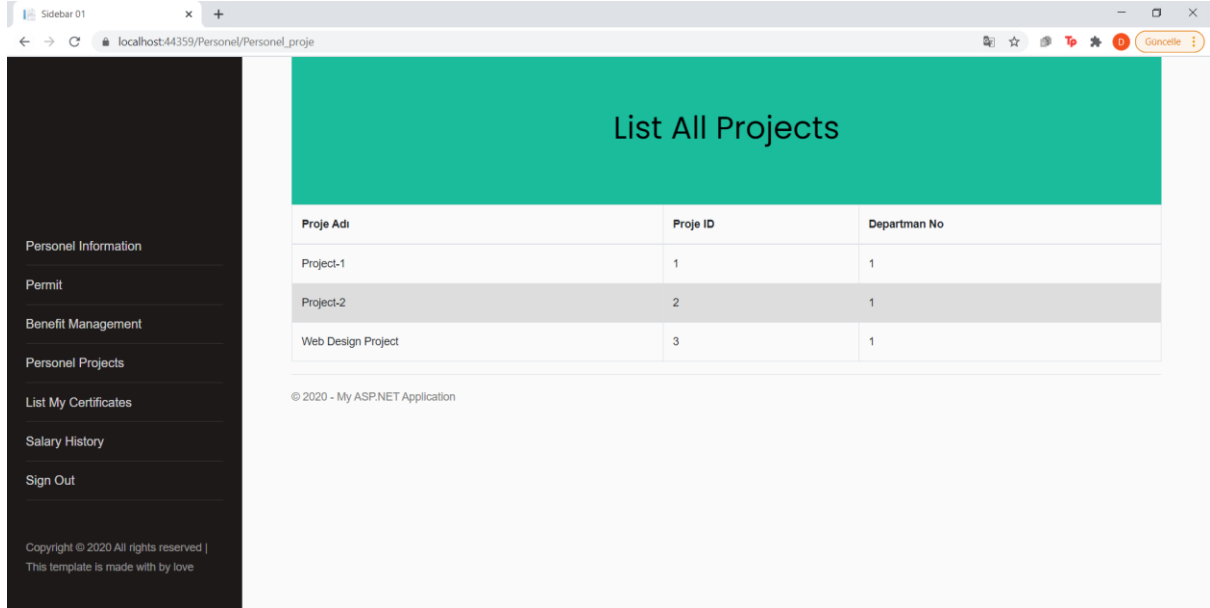
Salary History Ekranı

Kullanıcı sisteme giriş yapınca “Salary History” sekmesinden kendi geçmiş maaş bilgilerini görebileceği ekrana giriş yapar. Buradan admin tarafından kendisine ödemesi yapılmış ve sisteme girilmiş maaşlarını görüntüleyebilir. Ekranın sol tarafında yer alan butonlardan kendisiyle ilgili diğer sayfalara geçiş yapılabilir.

SQL Komutu

```
SqlDataAdapter sqlDA = new SqlDataAdapter("SELECT * FROM SalaryModel", sqlCon);  
var person = DbContext.Personel.Where(x => x.ID == id).FirstOrDefault();  
var salary = DbContext.SalaryModels.Where(x => x.pid == person.ID).FirstOrDefault();
```


6. Personelin Çalıştığı Projeleri Görüntüleme Ekranı



Personel Projects Ekranı

Kullanıcı sisteme giriş yapınca “Personel Projects” sekmesinden kendi çalıştığı projeleri ve bu projelerin proje bilgilerini görebileceği ekrana giriş yapar. Ekranın sol tarafında yer alan butonlardan kendiyle ilgili diğer sayfalara geçiş yapabilir.

SQL Komutu

```
List<Works_onModel> workson = DbContext.Works_onModels.Where(x=>x.personel_id==id).ToList();
```

```
List<ProjectModel> project = DbContext.ProjectModels.ToList();
```

```
List<ProjectModel> newproject = new List<ProjectModel> ();
```

```
foreach (var value in workson)
```

```
{
```

```
    newproject.Add(project.Where(x=>x.pid==value.project_id).FirstOrDefault());
```

```
    viewModel.ProjectModels.Add((project.Where(x => x.pid ==
```

```
value.project_id).FirstOrDefault()));
```

```
}
```

7. Personel İzin Ekranı

Personel Information

Permit

Benefit Management

Personel Projects

List My Certificates

Salary History

Sign Out

Permits

İzin Ekle

Search...

#	Açıklama	Başlangıç Tarihi	Bitiş Tarihi	Gün Sayısı	Onay	
1	Deneme 3	01/05/2021 00:00:00	01/10/2021 00:00:00	5	✓	<div>✎</div> <div>✖</div>
2	Saglık izni	20/12/2020 00:00:00	30/12/2020 00:00:00	10	✓	<div>✎</div> <div>✖</div>
3	Yillik Izin	15/01/2021 00:00:00	20/01/2021 00:00:00	5	×	<div>✎</div> <div>✖</div>

© 2020 - My ASP.NET Application

Personel İzin Tablosu

Kullanıcı sisteme Personel olarak giriş yaptıktan sonra sol taraftaki navigation barda permit (izin) sekmesine tıklayarak bu sayfaya ulaşabilir. Bu tabloda daha önceden kendi üzerine tanımlanmış izinleri açıklama, başlangıç/bitiş tarihleri, gün sayısı ve admin onayı olup olmadığını görebilmektedir. Eğer tablodaki izni admin tarafından onaylanmış ise “✓” onaylanmamış ise “×” ikonu ile temsil edilir. Kullanıcı tablonun sağ üst tarafında bulunan arama kutusu ile izinleri arasında arama yapabilmektedir. Her izin için bulunan kırmızı silme tuşu ile mevcut izni silebilir.

Kullanıcı Her izin satırında bulunan sarı düzenleme tuşuna basarak mevcut izni düzenleme ekranına erişerek o izne ait bilgileri güncelleyebilir.

Kullanıcı tablonun sol üstüne bulunan turkuaz “İzin Ekle” butonuna tıklayarak kendi üzerine yeni bir izin tanımlayabilir.

Hem düzenleme hem ekleme işlemleri sonrası admin onayı ihtiyacı olur ve onay “×” şeklinde güncellenir.

Düzenle

aciklama

Yillik Izin

baslangic_t

15/01/2021 00:00:00

bitis_t

20/01/2021 00:00:00

Save

[Back to List](#)

İzin Düzenleme Ekranı

İzin düzenleme ekranında kullanıcı mevcut bilgileri istediği şekilde değiştirdikten sonra altındaki turuncu “save” tuşuna basarak bilgileri güncelleyebilir. İşlemden vazgeçilmesi durumunda sayfanın en altında yer alan “Back to List” tuşu ile tekrar izinler tablosuna geri dönebilir.

Create

aciklama

baslangic_t

01/01/0001 00:00:00

bitis_t

01/01/0001 00:00:00

Create

Back to List

İzin Ekleme Ekranı

İzin ekleme ekranında kullanıcı yeni eklemek istediği iznin bilgilerini girdikten sonra altındaki turuncu “create” tuşuna basarak kendi üzerine yeni bir izin tanımlayabilir. İşlemden vazgeçilmesi durumunda sayfanın en altında yer alan “Back to List” tuşu ile tekrar izinler tablosuna geri dönebilir.

SQL SORGULARI:

Tablo oluşturma:

```
int toplamGun = (int)(Izin.bitis_t - Izin.baslangic_t).TotalDays;
using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    string query = "INSERT INTO [dbo].[IzinModel] VALUES( " + LoggedUserID +
", " + Izin.aciklama + ", " +
Izin.baslangic_t.ToString("yyyy-MM-dd") + ", " + Izin.bitis_t.ToString("yyyy-MM-dd") + ", " + toplamGun + ", " + 0 + ")";
    SqlCommand sqlCommand = new SqlCommand(query, sqlCon);
    sqlCommand.ExecuteNonQuery();
}
```

Delete:

```
using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    string query = "DELETE FROM [dbo].[IzinModel] WHERE ID = '" + ID + "'";
    SqlCommand sqlCommand = new SqlCommand(query, sqlCon);
    sqlCommand.ExecuteNonQuery();
}
```

Edit (Ekran Oluşturma):

```
using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    string query = "SELECT * FROM [dbo].[IzinModel] WHERE ID = '" + ID + "'";
    SqlDataAdapter sqlDa = new SqlDataAdapter(query, sqlCon);
    sqlDa.Fill(dtable);
    if (dtable.Rows.Count == 1)
    {
        Izin.ID = (int)dtable.Rows[0][0];
        Izin.pid = (int)dtable.Rows[0][1];
        Izin.aciklama = dtable.Rows[0][2].ToString();
        Izin.baslangic_t = Convert.ToDateTime(dtable.Rows[0][3].ToString());
        Izin.bitis_t = Convert.ToDateTime(dtable.Rows[0][4].ToString());
        Izin.gun_sayisi = (int)dtable.Rows[0][5];
        Izin.isApprove = (int)dtable.Rows[0][6];
        return View(Izin);
    }
}
```

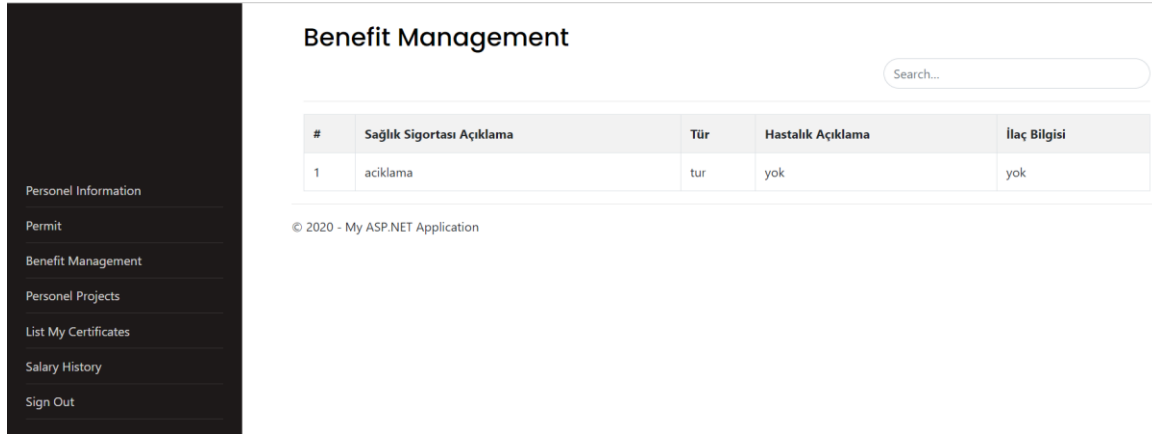
Edit(Update):

```
int toplamGun = (int)(Izin.bitis_t - Izin.baslangic_t).TotalDays;
using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    string query = "UPDATE [dbo].[IzinModel] SET aciklama = '" + Izin.aciklama +
    "', baslangic_t = '" + Izin.baslangic_t.ToString("yyyy-MM-dd") + "', bitis_t = '" +
    Izin.bitis_t.ToString("yyyy-MM-dd") + "', gun_sayisi = " + toplamGun + ", isApprove = " +
    0 + " WHERE ID = " + Izin.ID;
    SqlCommand sqlCommand = new SqlCommand(query, sqlCon);
    sqlCommand.ExecuteNonQuery();
}
```

Create:

```
int toplamGun = (int)(Izin.bitis_t - Izin.baslangic_t).TotalDays;
using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    string query = "INSERT INTO [dbo].[IzinModel] VALUES( " + LoggedUserID +
    ", " + Izin.aciklama + ", " + Izin.baslangic_t.ToString("yyyy-MM-dd") + ", " +
    Izin.bitis_t.ToString("yyyy-MM-dd") + ", " + toplamGun + ", " + 0 + ")";
    SqlCommand sqlCommand = new SqlCommand(query, sqlCon);
    sqlCommand.ExecuteNonQuery();
}
```

8. Personel Benefit Management Ekranı:



Personel Benefit Management Ekranı

Kullanıcı sisteme personel olarak giriş yaptıktan sonra ekranın yan tarafındaki navigation barda bulunan Benefit Management sekmesine tıklayarak bu sayfaya erişebilir. Bu sayfada kullanıcı admin tarafından kendi üstüne tanımlanmış olan bütün benefit management leri görüntüleyebilir. Tablonun sağ üstünde bulunan arama kutusu yardımı ile tablo içinde arama gerçekleştirilebilmektedir.

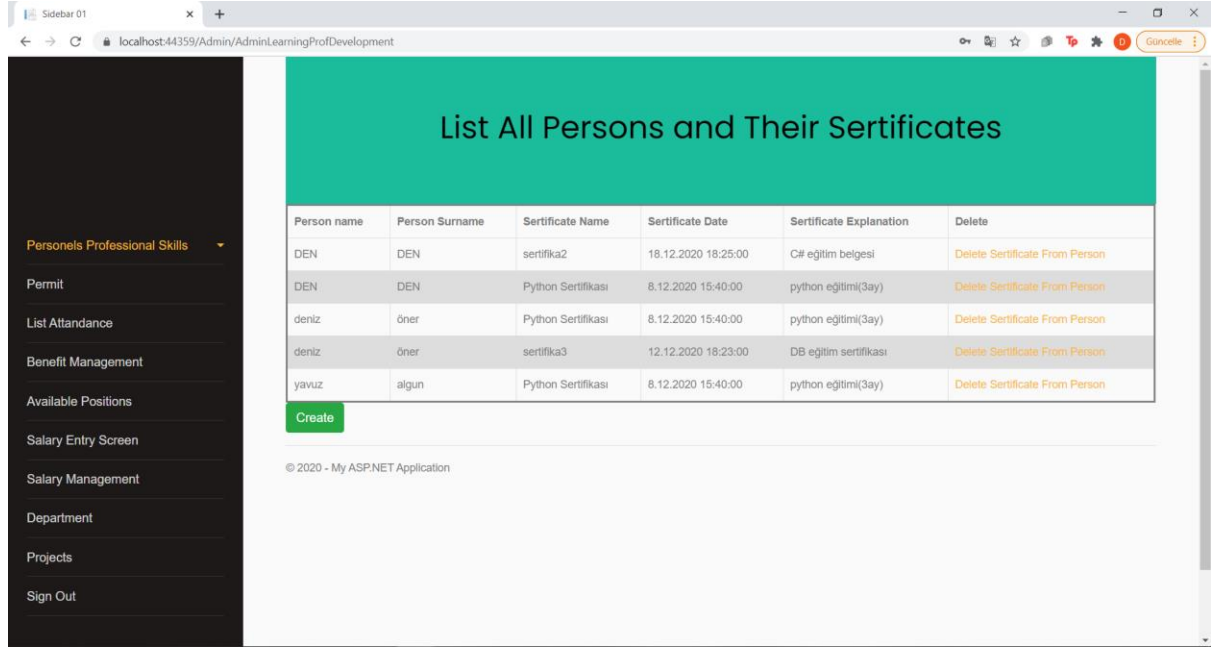
SQL KOMUTLARI:

```
DataTable dataTable = new DataTable();
using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    SqlDataAdapter sqlDA = new SqlDataAdapter("SELECT * FROM
[dbo].[Benef_ManModel] WHERE pid = '" + (int)LoggedUserID + "'", sqlCon);
    sqlDA.Fill(dataTable);
}

return View(dataTable);
```

Admin Girişi

1. Personellere ait Sertifika Görüntüleme Ekranı



Person name	Person Surname	Certificate Name	Certificate Date	Certificate Explanation	Delete
DEN	DEN	sertifika2	18.12.2020 18:25:00	C# eğitim belgesi	Delete Certificate From Person
DEN	DEN	Python Sertifikası	8.12.2020 15:40:00	python eğitimi(3ay)	Delete Certificate From Person
deniz	öner	Python Sertifikası	8.12.2020 15:40:00	python eğitimi(3ay)	Delete Certificate From Person
deniz	öner	sertifika3	12.12.2020 18:23:00	DB eğitim sertifikası	Delete Certificate From Person
yavuz	algün	Python Sertifikası	8.12.2020 15:40:00	python eğitimi(3ay)	Delete Certificate From Person

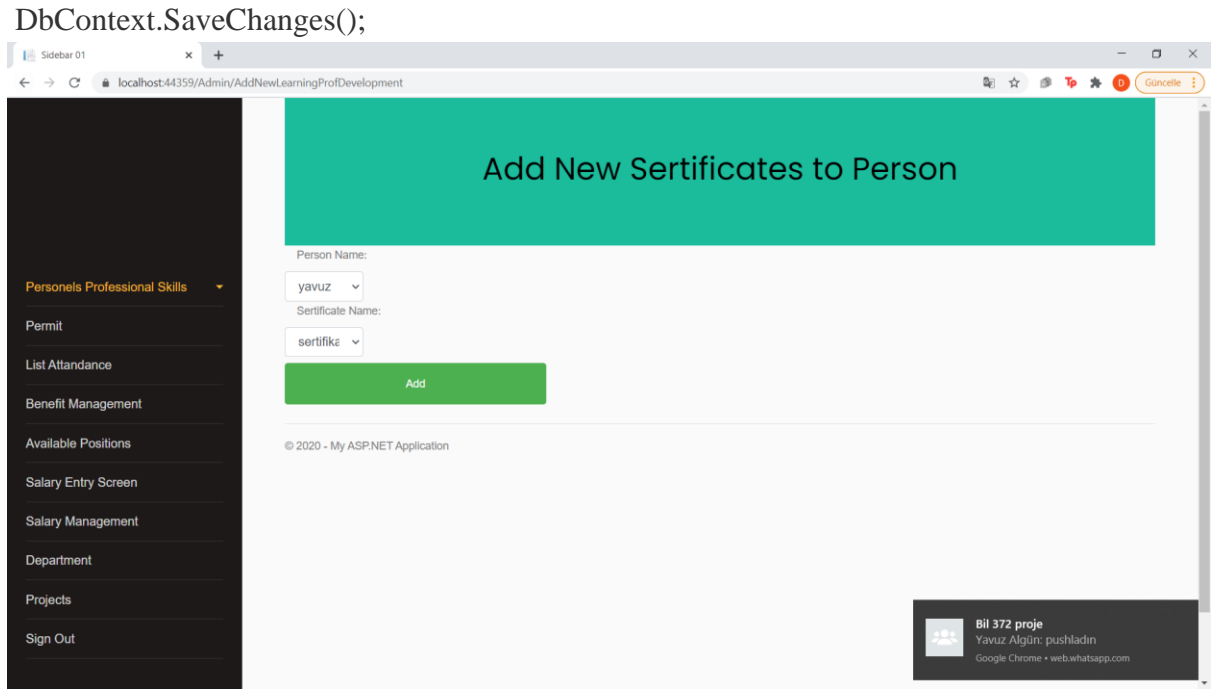
Personels Professional Skills Sayfası

Admin giriş yaptıktan sonra Personels Professional Skills sekmesinde sistemde yer alan sertifikaya sahip tüm kullanıcıları görüntüleyebilir. Create butonuna basarak bir kullanıcıya sistemde kayıtlı olan sertifika atama işlemini yapabilir, delete butonuna basarak kayıtlı veriyi sistemden silebilir.

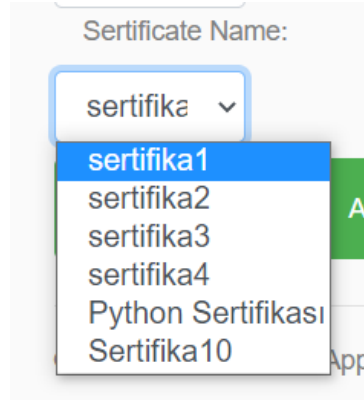
SQL KOMUTLARI

```
var viewModel = new PersonelViewModel();
viewModel.Personel = DbContext.Personel.ToList();
var Ldevelopment = DbContext.Learn_DevelopmentModels.ToList();
var LD = DbContext.Learn_DevelopmentModels.Include(c =>
c.Person).Include(c=>c.Sertifika).ToList();
ViewBag.Persons = new SelectList(DbContext.Personel.ToList(), "ID", "PersonName",
"PersonSurname");
ViewBag.Certificate = new SelectList(DbContext.SertifikaModels.ToList(), "ID",
"sertifika_adi");
DbContext.Learn_DevelopmentModels.Add(new Learn_DevelopmentModel
{
    person_id = pId,
    sertifika_id = sId
});
DbContext.SaveChanges();
DbContext.Learn_DevelopmentModels.Remove(DbContext.Learn_DevelopmentModels
.Find(id));
```

DbContext.SaveChanges();



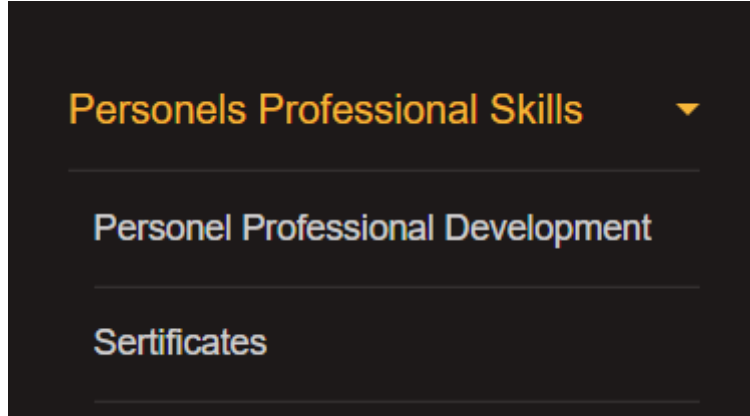
Add new Certificate to Person Ekran Görüntüsü



Create new Certificate to Person Ekranının pop-up görüntüsü

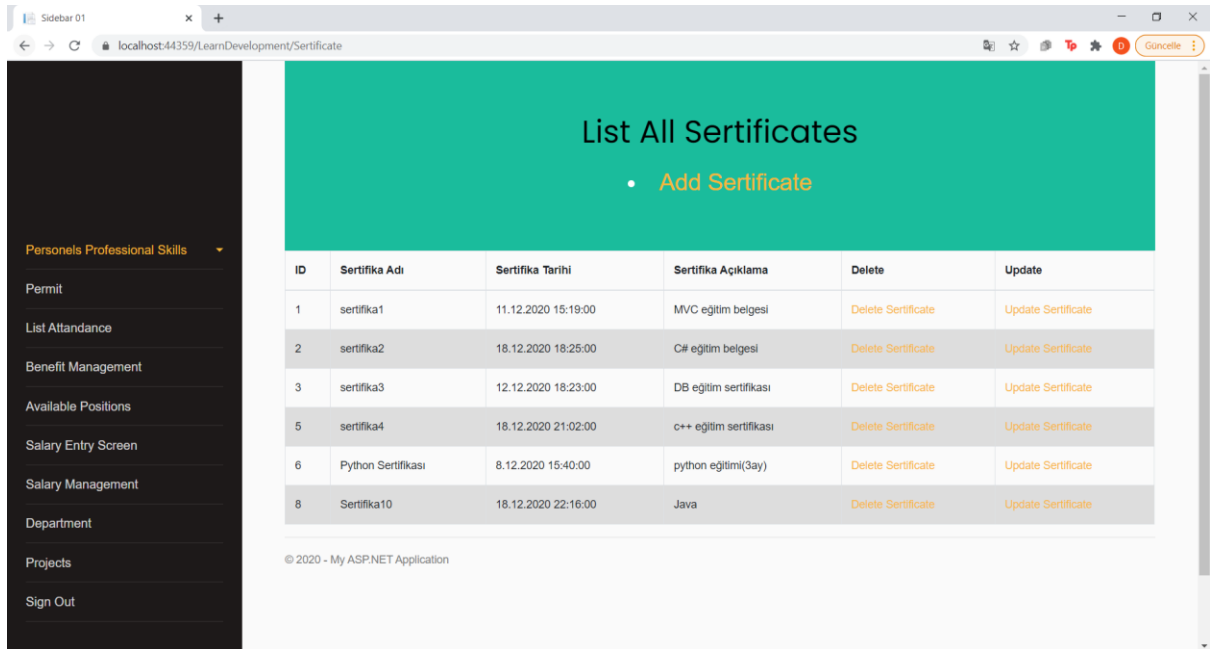
Kullanıcı personele yeni sertifika ekleme kısmından sistemde yer alan tüm sertifikaları görüntüleyip buradan istediğini seçebilir.

2.Sertifika Görüntüleme Ekranı



Layout Ekran Görüntüsü

Sertifikaları görüntülemek için admin personels Professional Skills sekmesinden “Certificate” kısmına tıklayarak Sertifika görüntüleme ekranına giriş yapabilir.



List All Certificates Ekran Görüntüsü

Admin giriş yaptıktan sonra Certificate sekmesinde sistemde yer alan sertifikaları görüntüleyebilir. “Add Certificate” butonuna basarak yeni bir sertifika oluşturabilir, “Delete Certificate” butonuna ilgili sertifikayı silebilir veya “Update Certificate” butonuna basarak ilgili sertifikayı güncelleyebilir.

Add New Sertificate ekran görüntüsü

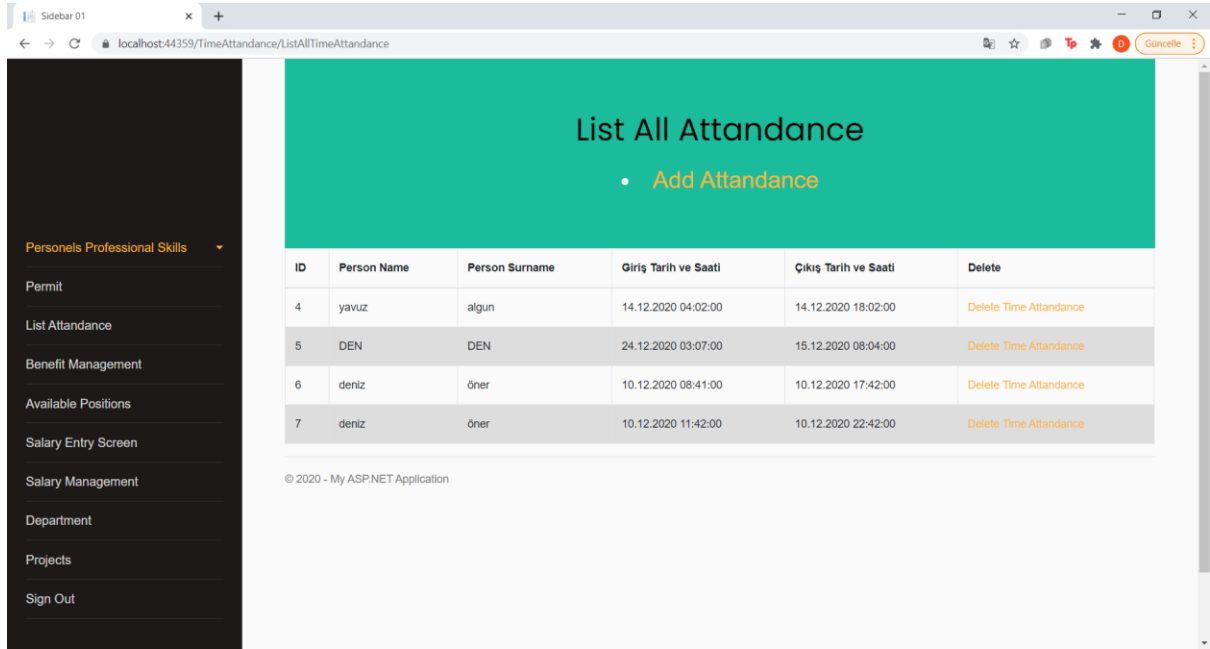
Update Sertifika Ekran Görüntüsü

SQL Komutları:

```
var viewModel = new SertifikaViewModel();  
viewModel.SertifikaModels = DbContext.SertifikaModels.ToList();  
DbContext.SertifikaModels.Add(sertifikamodel);  
DbContext.SaveChanges();  
DbContext.SertifikaModels.Remove(DbContext.SertifikaModels.Find(id));  
DbContext.SaveChanges();  
var ktg = DbContext.SertifikaModels.Find(sertifikaModel.ID);
```

```
ktg.sertifika_aciklama = sertifikaModel.sertifika_aciklama;
ktg.sertifika_adi = sertifikaModel.sertifika_adi;
ktg.sertifika_tarihi = sertifikaModel.sertifika_tarihi;
DbContext.SaveChanges();
var ktgr = DbContext.SertifikaModels.Find(id);
ViewData["Message"]=ktgr;
```

3.Attendance Görüntüleme Ekranı



ID	Person Name	Person Surname	Giriş Tarih ve Saati	Çıkış Tarih ve Saati	Delete
4	yavuz	algun	14.12.2020 04:02:00	14.12.2020 18:02:00	Delete Time Attendance
5	DEN	DEN	24.12.2020 03:07:00	15.12.2020 08:04:00	Delete Time Attendance
6	deniz	öner	10.12.2020 08:41:00	10.12.2020 17:42:00	Delete Time Attendance
7	deniz	öner	10.12.2020 11:42:00	10.12.2020 22:42:00	Delete Time Attendance

List All Attendance Ekranı

Admin giriş yaptıktan sonra “List Attendance” sekmesinde sistemde yer alan personellerin giriş çıkış saatlerini görüntüleyebilir. “Add Attendance” butonuna basarak yeni bir attendance oluşturabilir, “Delete Attendance” butonuna basarak ilgili attendance’ı silebilir.

Add New Attendance to Person ekranı

Admin sistemde yer alan tüm kullanıcıları Person Name combobox'ından görüntüleyebilir.

Giriş çıkış saati ekran görüntüsü

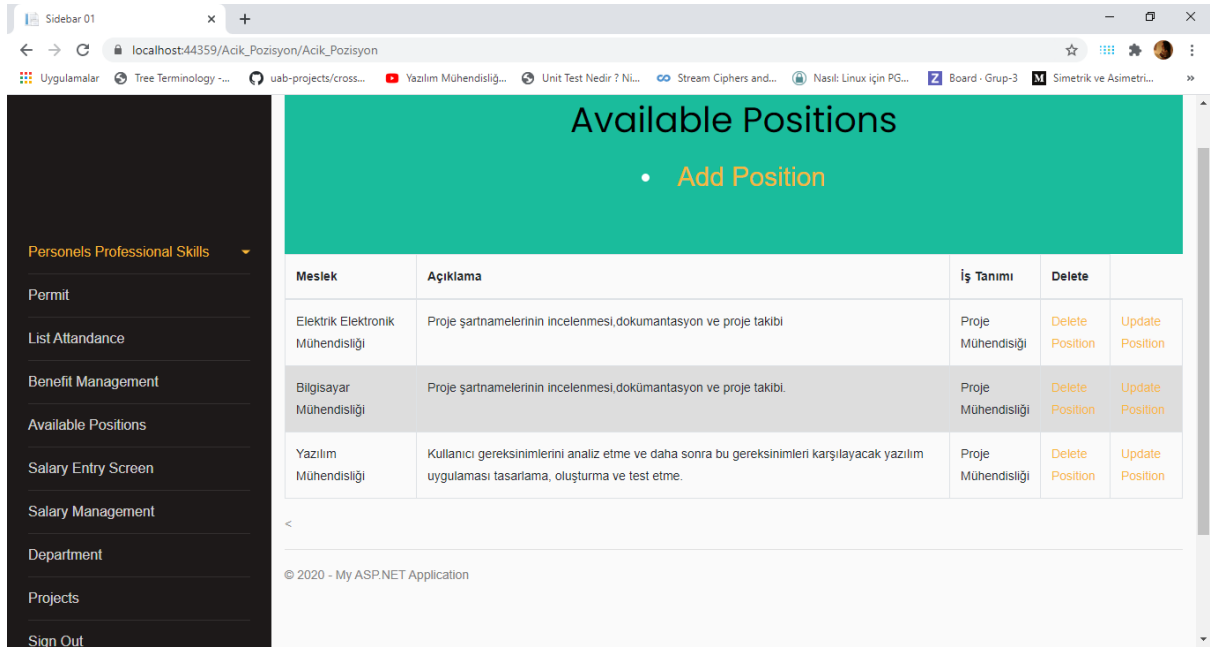
Admin sistemde yeni bir attendance eklediğinde çıkan pop-updan istediği gün ve saat bilgisini seçebilir.

SQL KOMUTLARI

```
var viewModel = new Time_AttendanceViewModel();
```

```
viewModel.Time_att = DbContext.Time_AttendanceModels.ToList();
ViewData["Message"] = DbContext.Time_AttendanceModels.ToList();
var LD = DbContext.Time_AttendanceModels.Include(c => c.PersonModel).ToList();
DbContext.Time_AttendanceModels.Remove(DbContext.Time_AttendanceModels.Find(id))
DbContext.SaveChanges();
var person = Request.Form["Persons"];
int pId = Int16.Parse(person);
DbContext.Time_AttendanceModels.Add(new Time_AttendanceModel
{
    person_id = pId,
    giris_saati = time_AttendanceModel.giris_saati,
    cikis_saati= time_AttendanceModel.cikis_saati
});
DbContext.SaveChanges();
```

4.Available Positions Ekranı



Meslek	Açıklama	İş Tanımı	Delete	Update
Elektrik Elektronik Mühendisliği	Proje şartnamelerinin incelenmesi,dokumantasyon ve proje takibi	Proje Mühendisliği	Delete Position	Update Position
Bilgisayar Mühendisliği	Proje şartnamelerinin incelenmesi,dokumantasyon ve proje takibi.	Proje Mühendisliği	Delete Position	Update Position
Yazılım Mühendisliği	Kullanıcı gereksinimlerini analiz etme ve daha sonra bu gereksinimleri karşılayacak yazılım uygulaması tasarlama, oluşturma ve test etme.	Proje Mühendisliği	Delete Position	Update Position

Available Position Ekranı

Admin giriş yaptıktan sonra “Available Positions”sekmesinde firmadaki uygun iş pozisyonları görüntüleyebilir ve yeni iş ilanı için uygun pozisyonu yaratabilir. “Add Position” butonuna basarak yeni bir iş pozisyonu oluşturabilir, “Delete Position” butonuna basarak ilgili position’ı silebilir ve “Update Position” butonuna basarak mevcut pozisyon tanımını değiştirip,güncelleyebilir.

The screenshot shows a web browser window with the URL `localhost:44359/Acik_Pozisyon/AddPosition`. The page has a dark sidebar on the left with the following menu items: "Personels Professional Skills" (expanded), "Permit", "List Attendance", "Benefit Management", "Available Positions", "Salary Entry Screen", and "Sign Out". The main content area has a teal header with the text "Add New Positions". Below the header, there are three input fields labeled "Meslek :", "Açıklama:", and "İş Tanımı:". A green button labeled "AddPosition" is at the bottom of the form. The footer of the page says "© 2020 - My ASP.NET Application".

Add New Positions Ekranı

The screenshot shows a web browser window with the URL `localhost:44359/Acik_Pozisyon/Acik_PozisyonGetir/9`. The page has a dark sidebar on the left with the following menu items: "Personels Professional Skills" (expanded), "Permit", "List Attendance", "Benefit Management", "Available Positions", "Salary Entry Screen", and "Sign Out". The main content area has a teal header with the text "Update Positions". Below the header, there are three input fields labeled "Meslek :", "Açıklama:", and "İş Tanımı:". The "Meslek :" field contains the text "Yazılım Mühendisliği". The "Açıklama:" field contains the text "Kullanıcı gereksinimlerini analiz etme ve daha sonra bu gereksinimleri karşılayacak yazılım uygulaması tasarlama, oluşturma ve test etme.". The "İş Tanımı:" field contains the text "Proje Mühendisliği". A green button labeled "Update" is at the bottom of the form. The footer of the page says "© 2020 - My ASP.NET Application".

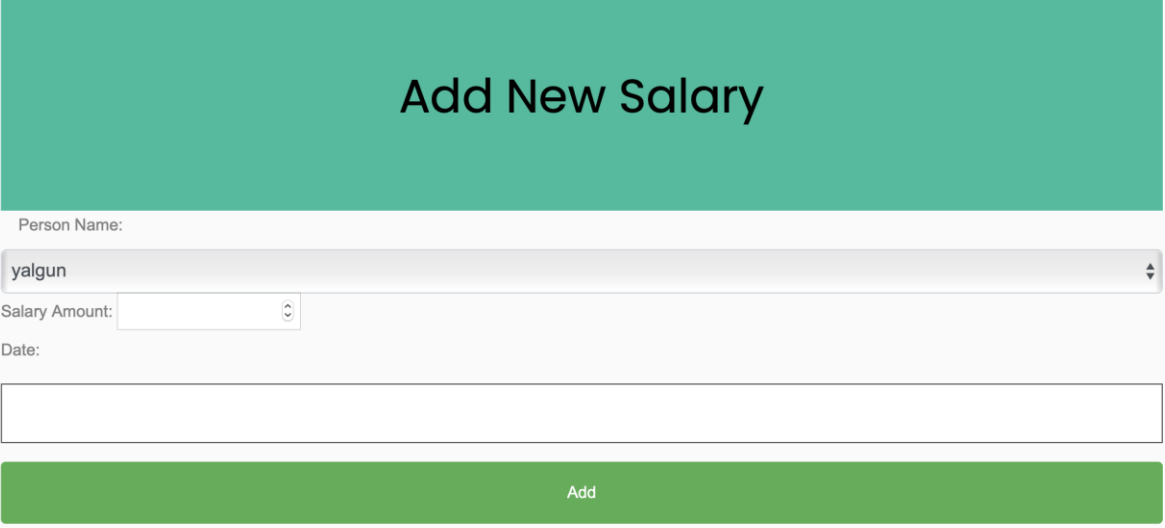
Update Positions Ekranı

```
var viewModel = new Acik_PozisyonViewModel();
viewModel.Acik_PozisyonModels = DbContext.Acik_PozisyonModels.ToList();
DbContext.Acik_PozisyonModels.Remove(DbContext.Acik_PozisyonModels.Find(id));

DbContext.Acik_PozisyonModels.Add(acikpozisyonmodel);

var ktgr = DbContext.Acik_PozisyonModels.Find(id);
var ktg = DbContext.Acik_PozisyonModels.Find(acikpozisyonModel.ID);
ktg.meslek = acikpozisyonModel.meslek;
    ktg.açıklama = acikpozisyonModel.açıklama;
    ktg.iş_tanımı = acikpozisyonModel.iş_tanımı;
    DbContext.SaveChanges();
```

5. Salary Entry Screen Ekranı



Person Name:

yalgun

Salary Amount:

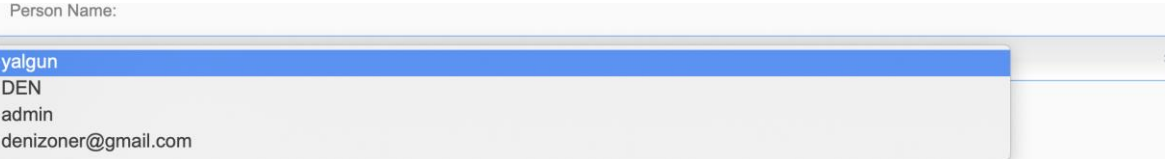
Date:

Add

© 2020 - My ASP.NET Application

Salary Entry Screen Ekran Görüntüsü

Admin giriş yaptıktan sonra Salary Entry Screen sekmesinden istediği bir kullanıcının maaş girişini yapabilir. Burada web uygulamasının user-friendly olabilmesi açısından sistemde kayıtlı olan ve adminin yetkisi altında çalışan kullanıcılar arasından maaş bilgisini girmek istediği kullanıcıyı aşağıdaki gibi dropdown yardımıyla kullanıcı rahatlıkla seçebilmektedir.



Person Name:

✓ yalgun

DEN

admin

denizoner@gmail.com

SQL KOMUTLARI

```
viewModel.SalaryModels = DbContext.SalaryModels.ToList();  
ViewData["Message"] = DbContext.SalaryModels.ToList();  
var LD = DbContext.SalaryModels.Include(c => c.PersonModel).ToList();
```

6. Projects Ekranı

List All Salary					
• Add Salary					
ID	Person Name	Person Surname	Salary Amount	Date	Delete
2	yavuz	algun	30000	1.01.2021 00:00:00	Delete Salary
2	yavuz	algun	28000	2.01.2021 00:00:00	Delete Salary
2	yavuz	algun	1200	12.12.2020 00:00:00	Delete Salary
3	DEN	DEN	20	12.12.2020 00:00:00	Delete Salary
3	DEN	DEN	30000	15.12.2020 00:00:00	Delete Salary

© 2020 - My ASP.NET Application

Salary Management Ekran Görüntüsü

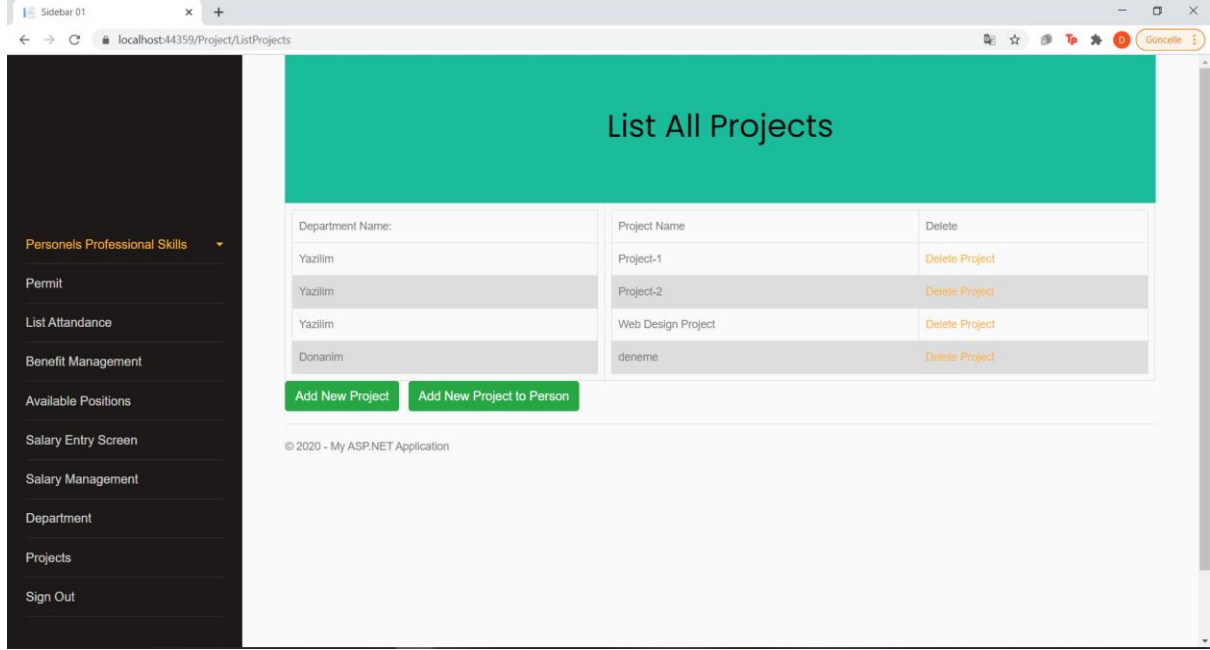
Admin giriş yaptıktan sonra Salary Management sekmesinden kendisine bağlı çalışan tüm kullanıcıların maaş bilgilerini görüntüleyebilmektedir. Eğer kullanıcı sayfada herhangi bir kullanıcı için maaş bilgisi girmek isterse Add Salary butonu üzerine basmalıdır. Buraya basıldığında otomatik olarak 4. madde de bahsedilen Salary Entry Screen sekmesine yönlendirilecektir. Eğer kullanıcı sisteme girilmiş bir maaş bilgisini silmek isterse delete salary butonuna basmalıdır.

SQL KOMUTLARI

```
ViewBag.Persons = new SelectList(DbContext.Personel.ToList(), "ID", "username",  
"PersonSurname");  
var person = Request.Form["Persons"];  
int pId = Int16.Parse(person);  
DbContext.SalaryModels.Add(new SalaryModel  
{pid = pId,miktar = salaryModel.miktar,tarih = salaryModel.tarih});  
DbContext.SaveChanges();  
var viewModel = new SalaryViewModel();  
viewModel.SalaryModels = DbContext.SalaryModels.ToList();
```

```
ViewData["Message"] = DbContext.SalaryModels.ToList();  
var LD = DbContext.SalaryModels.Include(c => c.PersonModel).ToList();  
DbContext.SalaryModels.Remove(DbContext.SalaryModels.Find(id));  
DbContext.SaveChanges();
```

7. Projects Ekranı



Projects Ekran Görüntüsü

Admin giriş yaptıktan sonra “Projects” sekmesinde firmadaki projeleri ve bağlı olduğu departmanları görüntüleyebilir, ”Add New Project” butonuna basarak proje yaratıp sistemdeki departmanlara bağlayabilir ve “Add New Project to Person” butonuna basarak sistemde kayıtlı olan projelere istenilen kişiler atanabilir. “Delete Project” butonuna basarak istenilen projeler silinebilir.

The screenshot shows a web browser window with the URL `localhost:44359/Project/AddNewProject`. The page has a dark sidebar on the left with a menu containing: 'Personels Professional Skills' (expanded), 'Permit', 'List Attendance', 'Benefit Management', 'Available Positions', 'Salary Entry Screen', 'Salary Management', 'Department', 'Projects', and 'Sign Out'. The main content area has a teal header with the text 'Add New Project'. Below the header, there is a form with a 'Proje Adı:' label and a text input field. Underneath, there is a 'Department Name:' label and a dropdown menu currently showing 'Yazılım'. A green 'Add' button is positioned below the dropdown. At the bottom of the form, there is a copyright notice: '© 2020 - My ASP.NET Application'. A 'Güncelle' button is visible in the top right corner of the browser window.

Add New Project Ekranı

The screenshot shows a web browser window with the URL `localhost:44359/Project/AddNewProjectToPerson`. The sidebar is identical to the previous screenshot. The main content area has a teal header with the text 'Add New Project to Person'. Below the header, there is a form with a 'Person Name:' label and a dropdown menu showing 'yavuz'. Below that, there is a 'Project Name:' label and a dropdown menu showing 'Project'. A green 'Add' button is positioned below the dropdowns. At the bottom of the form, there is a copyright notice: '© 2020 - My ASP.NET Application'. A 'Güncelle' button is visible in the top right corner of the browser window.

Add New Project to Person Ekranı

SQL KOMUTLARI

```
viewModel.Departments = DbContext.Departmes.ToList();
List<ProjectModel> project = DbContext.ProjectModels.ToList();
List<Department> newproject = new List<Department>();
dynamic mymodel = new ExpandoObject();
foreach (var value in project)
{
```

```
        int id = value.dno;
        newproject.Add(DbContext.Departmes.Find(id));
    }
mymodel.Project = project;
mymodel.Department = newproject;

var person = Request.Form["p_adı"];
var department = Request.Form["Department"];

int sId = Int16.Parse(department);
DbContext.ProjectModels.Add(new ProjectModel
{
    p_adı = person.ToString(),
    dno = sId
});
var person = Request.Form["Persons"];
var project = Request.Form["Project"];
int pId= Int16.Parse(person);

int sId = Int16.Parse(project);
DbContext.Works_onModels.Add(new Works_onModel
{
    personel_id = pId,
    project_id = sId
});
DbContext.SaveChanges();

DbContext.ProjectModels.Remove(DbContext.ProjectModels.Find(id));
DbContext.SaveChanges();
```

8. İzin Ekranı:

The screenshot shows the 'İzinler' (Leaves) screen in an admin interface. On the left is a dark sidebar with a menu. The main area has a title 'İzinler' and a search bar. Below is a table with 10 columns: #, İzin ID, Personel ID, Personel Ad Soyad, Açıklama, Başlangıç Tarihi, Bitiş Tarihi, Gün Sayısı, Onay, and a status icon. There are three rows of leave requests for 'yavuz algun'. The first row is 'Deneme 3' (Trial 3) from 01/05/2021 to 01/10/2021, 5 days, approved (checkmark). The second row is 'Sağlık izni' (Medical leave) from 25/12/2020 to 30/12/2020, 5 days, not approved (X). The third row is 'Yıllık İzin' (Annual leave) from 15/01/2021 to 20/01/2021, 5 days, not approved (X). At the bottom, there is a copyright notice: '© 2020 - My ASP.NET Application'.

#	İzin ID	Personel ID	Personel Ad Soyad	Açıklama	Başlangıç Tarihi	Bitiş Tarihi	Gün Sayısı	Onay	
1	3	2	yavuz algun	Deneme 3	01/05/2021 00:00:00	01/10/2021 00:00:00	5	✓	
2	13	2	yavuz algun	Sağlık izni	25/12/2020 00:00:00	30/12/2020 00:00:00	5	×	
3	14	2	yavuz algun	Yıllık İzin	15/01/2021 00:00:00	20/01/2021 00:00:00	5	×	

Admin İzin Ekranı

Kullanıcı sisteme admin olarak giriş yaptıktan sonra ekranın sol tarafında yer alan siyah navigation barda bulunan Permit sekmesine tıklayarak bu ekrana ulaşabilir. Admin bu ekranda sisteme daha önceden kaydedilmiş bütün personelin izinlerini detayları ile birlikte görüntüleyebilir. Personel yeni bir izin ekledikten ve mevcut bir izni güncelledikten sonra admin onayına ihtiyaç duymaktadır. Admin, her izin satırında bulunan yeşil onay butonunu kullanarak mevcut izni bu sayfadan onaylayabilir. Admin sağ üst tarafta bulunan arama kutusu yardımı ile tablo içerisinde arama gerçekleştirebilir.

SQL KOMUTLARI:

Tablo Oluşturma:

```
DataTable dataTable = new DataTable();
using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    SqlDataAdapter sqlDA = new SqlDataAdapter("SELECT a.ID, a.pid,
b.PersonName, b.PersonSurname, a.aciklama, a.baslangic_t, a.bitis_t, a.gun_sayisi,
a.isApprove " +
" FROM [dbo].[IzinModel] as a , [dbo].[Personel] as b WHERE a.pid = b.ID ",
sqlCon);
    sqlDA.Fill(dataTable);
}
```

İzini Onaylama:

```
using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    string query = "UPDATE [dbo].[IzinModel] SET isApprove = " + 1 + " WHERE
ID = " + ID + """;
    SqlCommand sqlCmd = new SqlCommand(query, sqlCon);
    sqlCmd.ExecuteNonQuery();
}
```

9. Benefit Management Ekranı:

Benefit Management

Hak Ekle

Search...

#	Personel Adı Soyadı	Sağlık Sigortası Açıklama	Tür	Hastalık Açıklama	İlaç Bilgisi	
1	yavuz algun	acıklama	tur	yok	yok	
2	DEN DEN	asd	asd	asda	asd	

© 2020 - My ASP.NET Application

Admin Benefit Management Ekranı

Kullanıcı sisteme admin olarak giriş yaptıktan sonra ekranın solunda bulunan siyah navigation bar içinde bulunan Benefit Management sekmesine tıklayarak bu sayfaya erişebilir. Admin bu sayfada sistemde bulunan bütün personellerin üzerine tanımlanmış Benefitleri ayrıntılı olarak görüntüleyebilir. Tablonun sağ üstünde bulunan arama kutusu yardımı ile tablo içinde arama yapabilmektedir.

Ekranın sol üstünde bulunan turkuaz Hak ekle butonu ile yeni bir Benefit eklenebilmesi için benefit ekleme sayfasına yönlendirmektedir.

Her benefit için satırlarda ayrı ayrı bulunan turuncu edit butonu Benefitleri düzenlemek için benefit düzenleme ekranına yönlendirir.

Her benefit için satırlarda ayrı ayrı bulunan kırmızı delete butonuna tıklandığında mevcut benefit silinir.

The screenshot shows the 'Create' form for 'Benef_ManModel'. On the left is a dark sidebar with a menu containing 'Personels Professional Skills', 'Permit', 'List Attendance', 'Benefit Management', 'Available Positions', 'Salary Entry Screen', 'Salary Management', 'Department', and 'Projects'. The main content area has a title 'Create' and 'Benef_ManModel'. Below this is a form with a dropdown for 'Person Username' showing 'yalgun' and a list of suggestions: 'yalgun', 'DEN', 'admin', and 'denizoner@gmail.com'. There are three text input fields: 'tur', 'hastalik_aciklama', and 'ilac_bilgisi'. At the bottom, there is an orange 'Create' button and a 'Back to List' link.

Benefit Create Ekranı

Admin Benefit ekleme ekranında combobox ile istediği kullanıcıyı (kullanıcı adı ile) seçebilmekte ve gerekli diğer bilgileri doldurduktan sonra sayfanın altında bulunan turuncu Create butonu ile yeni Benefit oluşturabilir. İşlemden vazgeçilmesi durumunda sayfanın en altında yer alan “Back to List” tuşu ile tekrar izinler tablosuna geri dönebilir.

The screenshot shows the 'Edit' form for 'Benef_ManModel'. The sidebar is the same as in the 'Create' form. The main content area has a title 'Edit' and 'Benef_ManModel'. The form includes a dropdown for 'Person UserName' with 'yalgun' selected. Below it are five text input fields: 'sag_sig_aciklama', 'aciklama', 'tur', 'hastalik_aciklama', and 'ilac_bilgisi'. At the bottom, there is an orange 'Save' button and a 'Back to List' link. A footer at the very bottom reads '© 2020 - My ASP.NET Application'.

Benefit Edit Ekranı

Admin Benefit edit ekranında girdiğinde editlemek istediği benefit management in mevcut bilgileri tabloya doldurulmuş olarak gelir. Kullanıcı kullanıcı adı combobox ta seçili olarak gelir istendiği takdirde yine combobox ile değiştirilebilir. Admin buradaki bilgileri istediği gibi değiştirdikten sonra sayfanın altında bulunan turuncu “Save” butonu ile Benefit bilgilerini güncelleyebilir. İşlemden vazgeçilmesi durumunda sayfanın en altında yer alan “Back to List” tuşu ile tekrar izinler tablosuna geri dönebilir.

SQL KOMUTLARI:

Sayfa Oluşturma:

```
DataTable dataTable = new DataTable();
using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    SqlDataAdapter sqlDA = new SqlDataAdapter("SELECT a.PersonName,
a.PersonSurname, b.sag_sig_aciklama, b.tur, b.hastalik_aciklama, b.ilac_bilgisi, b.ID FROM
[dbo].[Benef_ManModel] AS b, [dbo].[Personel] AS a WHERE b.pid = a.ID", sqlCon);
    sqlDA.Fill(dataTable);
}
```

Delete:

```
using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    string query = "DELETE FROM [dbo].[Benef_ManMode] WHERE ID = " + ID +
""";
    SqlCommand sqlCommand = new SqlCommand(query, sqlCon);
    sqlCommand.ExecuteNonQuery();
}
```

Edit (Tablo Oluşturma):

```
Benef_ManModel Izin = new Benef_ManModel();
DataTable dtable = new DataTable();

using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    string query = "SELECT * FROM [dbo].[Benef_ManModel] WHERE ID = " + ID
+ """;
    SqlDataAdapter sqlDa = new SqlDataAdapter(query, sqlCon);
    sqlDa.Fill(dtable);
    if (dtable.Rows.Count == 1)
    {
        Izin.ID = (int)dtable.Rows[0][0];
        Izin.pid = (int)dtable.Rows[0][1];
        Izin.sag_sig_aciklama = dtable.Rows[0][2].ToString();
        Izin.tur = dtable.Rows[0][3].ToString();
        Izin.hastalik_aciklama= dtable.Rows[0][4].ToString();
    }
}
```

```
        Izin.ilac_bilgisi= dtable.Rows[0][5].ToString();
        ViewBag.Persons = new SelectList(DbContext.Personel.ToList(), "ID",
"username", Izin.pid);
        return View(Izin);
    }
}
```

Edit(Güncelleme):

```
using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    string query = "UPDATE [dbo].[Benef_ManModel] SET pid = " +
BenefitManagement.pid + ", sag_sig_aciklama = " +
BenefitManagement.sag_sig_aciklama+ ", hastalik_aciklama = " +
BenefitManagement.hastalik_aciklama + ", tur = " + BenefitManagement.tur + " ilac_bilgisi
= " + BenefitManagement.ilac_bilgisi + " WHERE ID = " + BenefitManagement.ID;
    SqlCommand sqlCmd = new SqlCommand(query, sqlCon);
    sqlCmd.ExecuteNonQuery();
}
return RedirectToAction("IzinGiris");
}
```

Create:

```
using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    string query = "INSERT INTO [dbo].[Benef_ManModel] VALUES(" +
BenefitManagement.pid + ", " + BenefitManagement.sag_sig_aciklama +
    ", " + BenefitManagement.tur + ", " + BenefitManagement.hastalik_aciklama +
    ", " +
    BenefitManagement.ilac_bilgisi + ")";
    SqlCommand sqlCommand = new SqlCommand(query, sqlCon);
    sqlCommand.ExecuteNonQuery();
}
```

10. Department Ekranı

Department

[Departman Ekle](#)

#	Departman Ismi	Departman Adresi	
1	Yazilim	Ankara	
2	Donanim	Odemis	
3	Grafik Tasarimi	Ankara/Teknokent	

© 2020 - My ASP.NET Application

Department Ekranı

Sisteme admin olarak giriş yapan kullanıcı ekranın solunda yer alan siyah navigation bar içinde bulunan Department sekmesi ile bu sayfaya ulaşabilir. Bu ekranda sistem içerisinde yer alan bütün departmanların adının ve adresinin yer aldığı bir tablo bulunur. Ekranın sağ üstünde yer alan arama kutusu yardımı ile tablo içinde arama gerçekleştirilebilir. Ekranın sol üstünde yer alan turkuaz departman ekle butonu ile admin departmanın ekleneceği sayfaya yönlendirilir.

Her departman için satırlarda ayrı ayrı bulunan turuncu edit butonu Departmanları düzenlemek için benefit düzenleme ekranına yönlendirir.

Her departman için satırlarda ayrı ayrı bulunan kırmızı delete butonuna tıklandığında mevcut departman silinir.

Create Department

DepartmentName

DepartmentAddress

[Create](#)

[Back to List](#)

Departman Ekleme Ekranı

Admin Departman oluşturma ekranına geldiğinde departman ismi ve departman adresi bilgilerini doldurduktan sonra ekranın altında bulunan turuncu Create tuşuna basarak yeni bir departman ekleyebilir. İşlemden vazgeçilmesi durumunda sayfanın en altında yer alan “Back to List” tuşu ile tekrar izinler tablosuna geri dönebilir.

Edit Department

DepartmentName

Yazılım

DepartmentAdress

Ankara

Save

[Back to list](#)

Departman Güncelleme Ekranı

Admin Departmen güncelleme ekranına geldiğinde güncellemek istediği departmanın bilgileri halihazırda doldurulmuş olarak gelir. Admin istediği bilgileri değiştirdikten sonra ekranın altında bulunan turuncu Save butonuna tıklayarak mevcut departmanı güncelleyebilir. İşlemden vazgeçilmesi durumunda sayfanın en altında yer alan “Back to List” tuşu ile tekrar izinler tablosuna geri dönebilir.

SQL KOMUTLARI:

Tabloyu Oluşturma:

```
DataTable dataTable = new DataTable();
using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    SqlDataAdapter sqlDA = new SqlDataAdapter("SELECT * FROM
[dbo].[Department] ", sqlCon);
    sqlDA.Fill(dataTable);
}
```

Delete:

```
using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    string query = "DELETE FROM [dbo].[Department] WHERE DepartmentID = '" +
DepartmentID + "'";
    SqlCommand sqlCommand = new SqlCommand(query, sqlCon);
    sqlCommand.ExecuteNonQuery();
}
```

Edit (Sayfayı Doldurma):

```
Department department= new Department();
```

```
DataTable dtable = new DataTable();
using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    string query = "SELECT * FROM [dbo].[DepartmentT] WHERE DepartmentID ="
+ DepartmentID + """;
    SqlDataAdapter sqlDa = new SqlDataAdapter(query, sqlCon);
    sqlDa.Fill(dtable);
    if (dtable.Rows.Count == 1)
    {
        department.DepartmentID = (int)dtable.Rows[0][0];
        department.DepartmentName = dtable.Rows[0][1].ToString();
        department.DepartmentAdress = dtable.Rows[0][2].ToString();

        return View(department);
    }
}
```

Edit(Update):

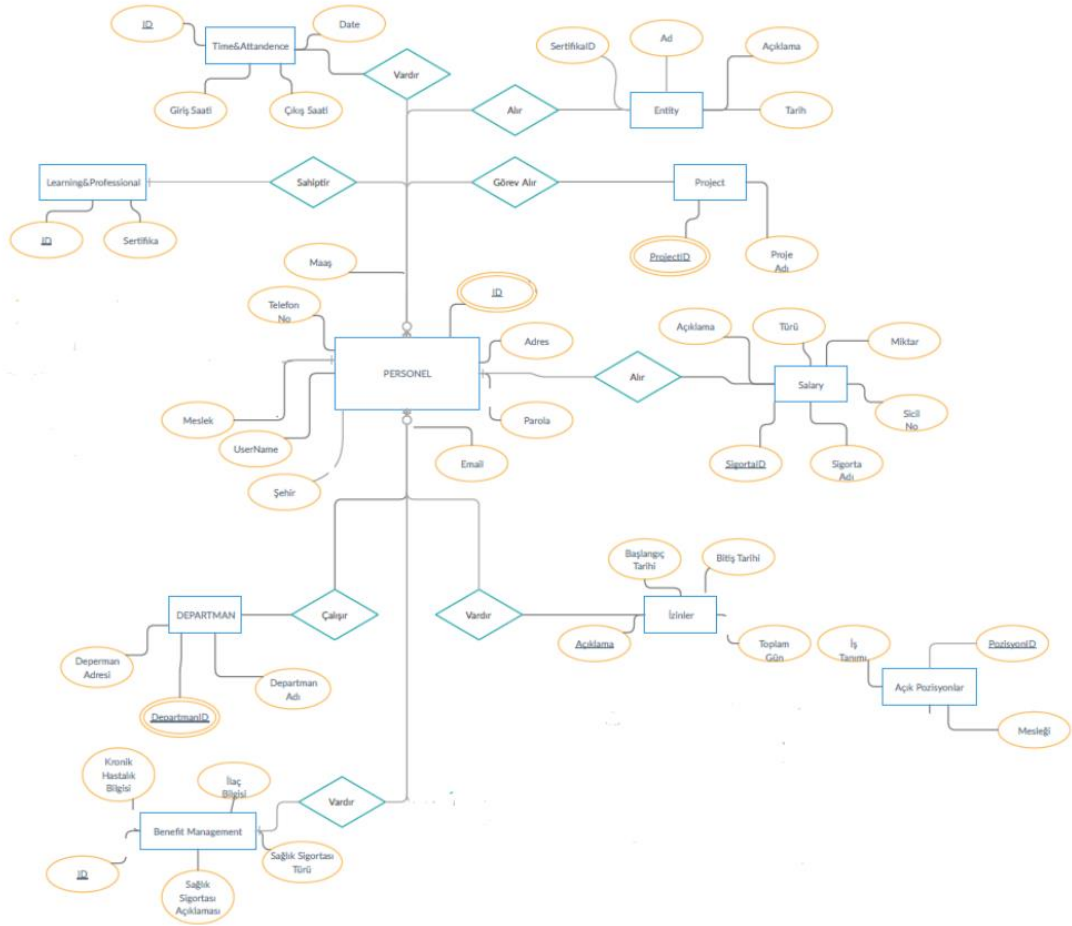
```
using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    string query = "UPDATE [dbo].[Department] SET DepartmentName = " +
department.DepartmentName + ", DepartmentAdress = " + department.DepartmentAdress +
"" WHERE DepartmentID = " + department.DepartmentID;
    SqlCommand sqlCmd = new SqlCommand(query, sqlCon);
    sqlCmd.ExecuteNonQuery();
}
return RedirectToAction("Index");
```

Create:

```
using (SqlConnection sqlCon = new SqlConnection(connectionString))
{
    sqlCon.Open();
    string query = "INSERT INTO [dbo].[Department] VALUES(" +
department.DepartmentName + ", " + department.DepartmentAdress + ")";
    SqlCommand sqlCommand = new SqlCommand(query, sqlCon);
    sqlCommand.ExecuteNonQuery();
}
```

EKLER

Projenin ER Diagramı



ER Diagramı ekran görüntüsü