

PROJE KONUSU: Bir pizza dükkanındaki gelen siparişler, müşteriler ve bunlara dair satış kayıtlarını tutacak olan bir veri tabanı projesi.

AD-SOYAD

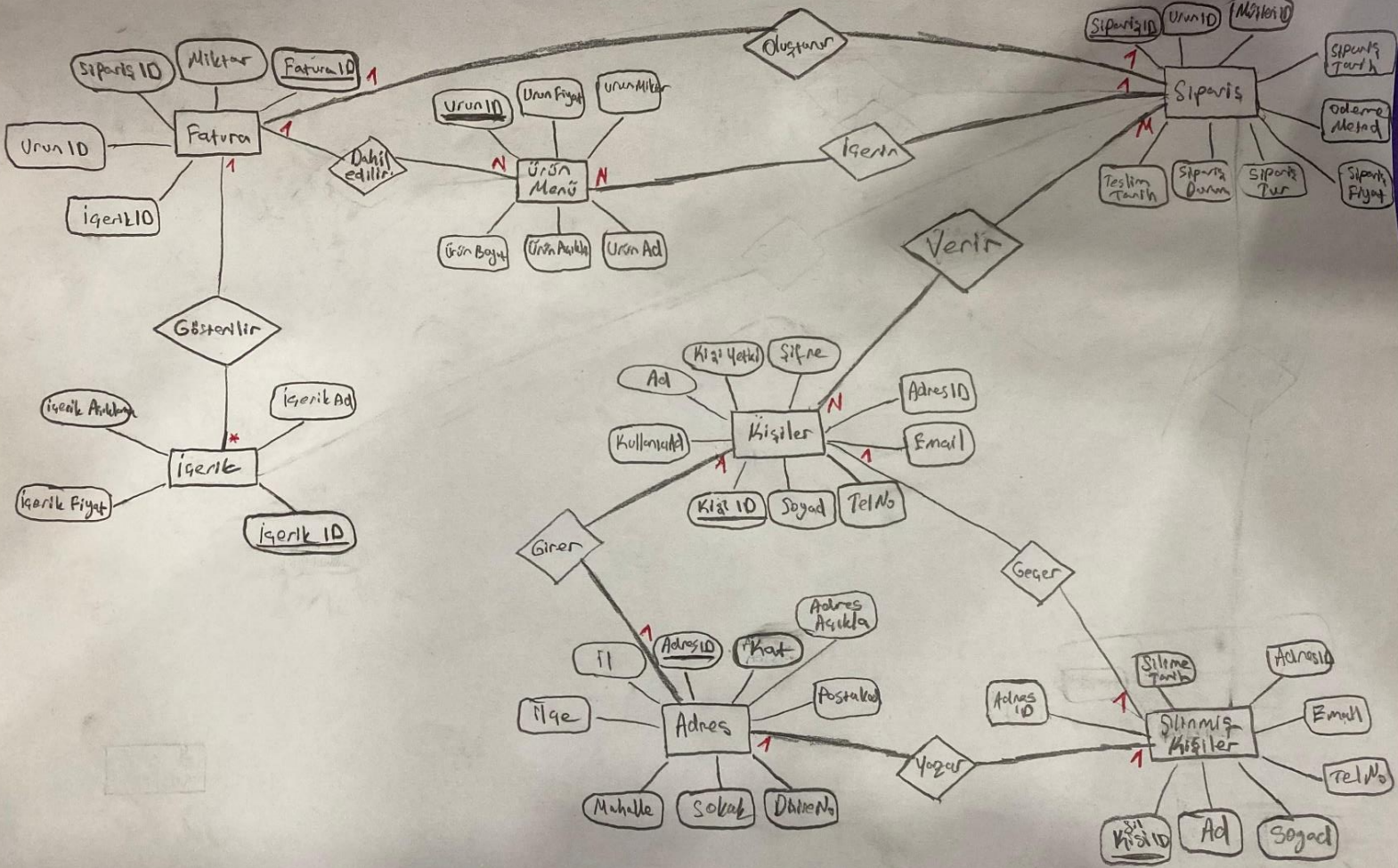
Erdem ASLAN

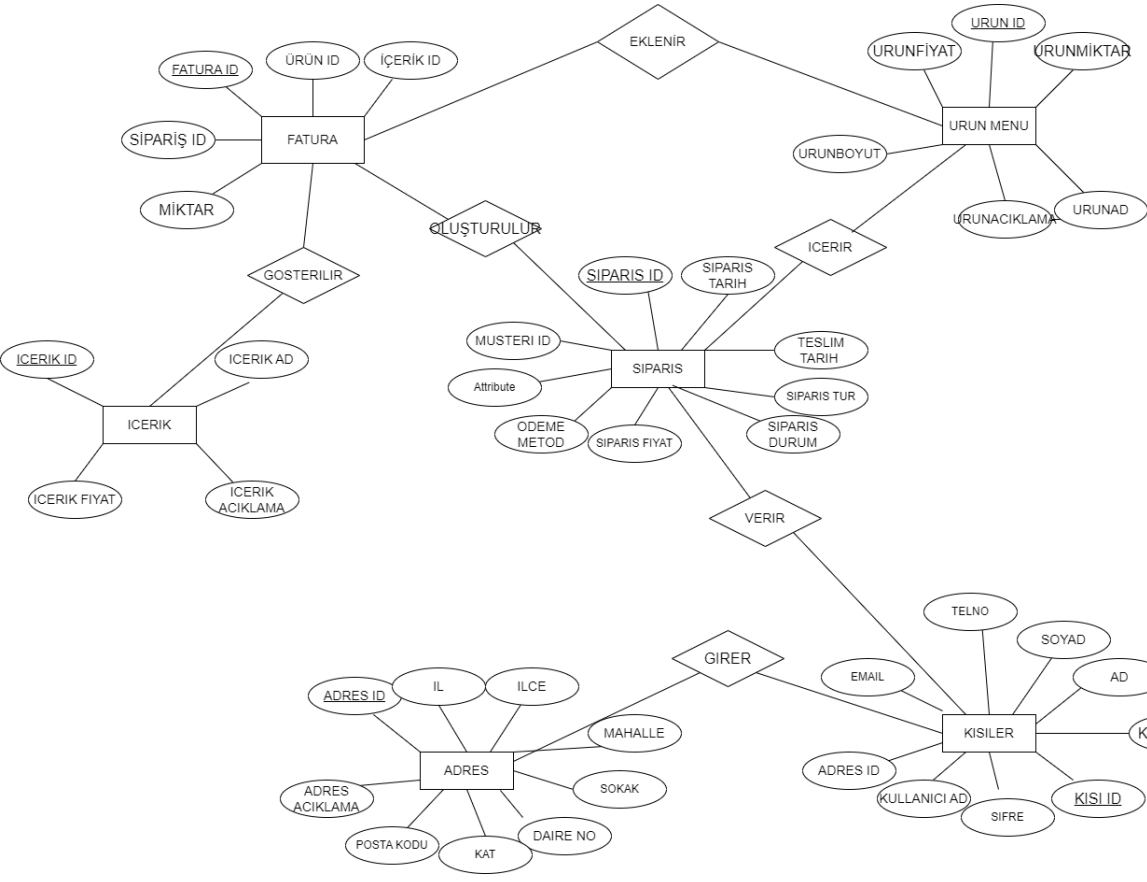
1. SENARYO

Bir pizza dükkanındaki gelen siparişler, müşteriler ve bunlara dair satış kayıtlarını tutacak olan bir veri tabanı projesi.

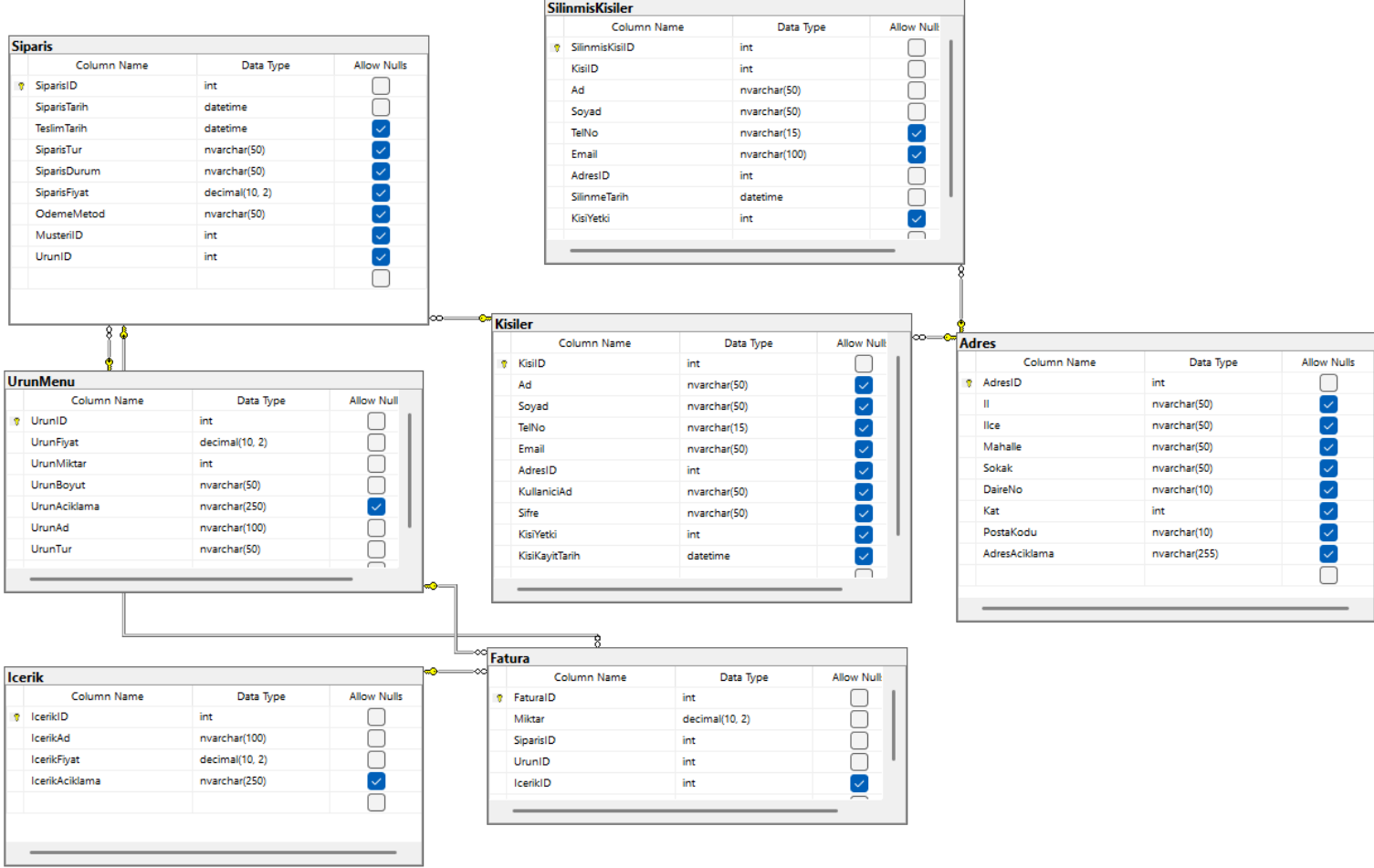
2. ER DİYAGRAMLARI

Pizzacı ER Diagram

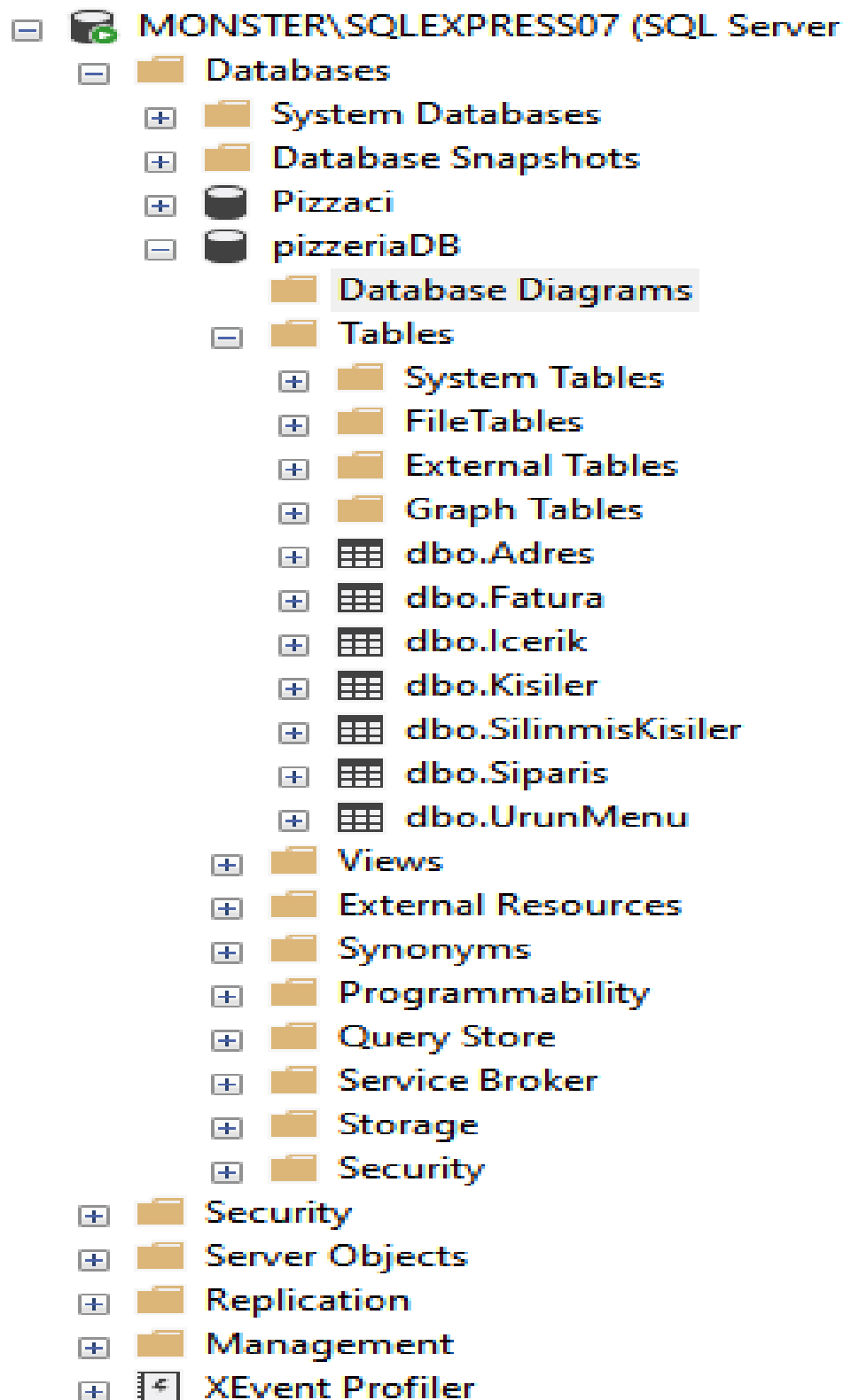




3. VERİTABANI ŞEMASI



4. SQL SERVER VERİTABANI



4.1 VERİTABANI TABLOLARI

1) Adres Tablosu

```
--PIZZACI ADINDA VERİTABANI OLUŞTURMA
```

```
Create Database pizzeriaDB
```

```
Use pizzeriaDB
```

```
--ADRES TABLOSU
```

```
Create Table Adres(  
  AdresID int Primary Key,  
  Il nvarchar(50),  
  Ilce nvarchar(50),  
  Mahalle nvarchar(50),  
  Sokak nvarchar(50),  
  DaireNo nvarchar(10),  
  Kat int,  
  PostaKodu int,  
  AdresAciklama nvarchar(255),  
)
```

2) Fatura Tablosu

```
--FATURA TABLOSU
```

```
Create Table Fatura(  
  FaturaID int Primary Key,  
  Tutar decimal (10,2),  
  SiparisID int Foreign Key references Siparis(SiparisID),  
  UrunID int Foreign Key references UrunMenu(UrunID),  
  IcerikID int Foreign Key references Icerik(IcerikID)  
)
```

3) İçerik Tablosu

```
--İÇERİK TABLOSU
```

```
Create Table Icerik(  
  IcerikID int Primary Key,  
  IcerikAd nvarchar(100),  
  IcerikAciklama nvarchar(250),  
  IcerikFiyat decimal(10,2),  
  Constraint ck_IcerikFiyat check (IcerikFiyat > 0)  
)
```

4) Kişiler Tablosu

--KİŞİLER TABLOSU

```
CREATE TABLE Kisiler(
    KisiID int identity (1,1) Primary Key,
    Ad nvarchar(50),
    Soyad nvarchar(50),
    TelNo nvarchar(15),
    Email nvarchar(50),
    AdresID int Foreign Key references Adres(AdresID),
    KullaniciAd nvarchar(50),
    Sifre nvarchar(50),
    KisiYetki int,
    KayitTarih datetime default getdate()
)
```

5) Silinmiş Kişiler Tablosu

--SİLİNMIŞ KİŞİLER TABLOSU

```
CREATE TABLE SilinmişKisiler(
    SilinmisKisiID int Primary Key,
    KisiID int Foreign Key references Kisiler(KisiID),
    Ad nvarchar(50),
    Soyad nvarchar(50),
    TelNo nvarchar(15),
    Email nvarchar(100),
    AdresID int Foreign Key references Adres(AdresID),
    SilinmeTarih datetime default getdate()
)
```

6) Sipariş Tablosu

--SİPARİŞ TABLOSU

```
Create Table Siparis(
    SiparisID int Primary Key,
    SiparisTarih datetime,
    TeslimTarih datetime,
    SiparisTur nvarchar(50),
    SiparisDurum nvarchar(50),
    SiparisFiyat decimal(10,2),
    OdemeMetod nvarchar(50),
    Constraint ck_OdemeMetod check (OdemeMetod in('Nakit','KrediKarti','BankaKarti')),
    KisiID int Foreign Key references Kisiler(KisiID),
    UrunID int Foreign Key references UrunMenu(UrunID),
    AdresID int Foreign Key references Adres(AdresID)
)
```

7) Ürün Menü Tablosu

```
--ÜRÜN MENÜLERİNİN TABLOSU
Create Table UrunMenu(
  UrunID int Primary Key,
  UrunFiyat decimal(10,2),
  UrunMiktar int,
  UrunBoyut nvarchar(50),
  UrunAciklama nvarchar(250),
  IcerikID int Foreign Key references Icerik(IcerikID),
  UrunAd nvarchar(100)
)
```

4.2 TRIGGERLAR

1)

```
CREATE TRIGGER trg_KisiDelete
ON SilinmisKisiler
  INSTEAD OF DELETE
AS
BEGIN
  SELECT 'Kayıt silinebilir.'
END
```

2)

```
CREATE TRIGGER trg_KisiInsert
ON Kisiler
  FOR INSERT
AS
BEGIN
  INSERT INTO Kisiler (KisiID, Ad, Soyad, TelNo, Email, AdresID, KullaniciAd, Sifre, KisiYetki)
  VALUES ('800', 'Ali', 'Veli', '1234567890', 'ali.veli@example.com', 1, 'ali123', 'sifre123', 'Kullanici');
END;
```


3)

```
CREATE TRIGGER trg_YeniSiparis
ON Siparis
AFTER INSERT
AS
BEGIN
    SELECT 'Yeni sipariş geldi.'
END
```

4)

```
CREATE TRIGGER trg_SiparisOnay
ON Siparis
AFTER INSERT
AS
BEGIN
    SELECT 'Sipariş alındı hazırlanıyor.'
END
```

5)

```
CREATE TRIGGER trg_S_aciklamakontrol
ON UrunMenu
AFTER INSERT
AS
BEGIN
    SELECT 'Girilen siparişte açıklama olup olmadığının kontrolünü yapın.'
END
```

6)

```
CREATE TRIGGER trg_Adres
ON Adres
AFTER INSERT
AS
BEGIN
    SELECT 'Yeni adres bilgisi geldi.'
END
```

4.3 STORED PROCEDURES

1) Kişi Ekleme Prosedürü

```
GO
CREATE PROCEDURE sp_InsertKisi --1 KİŞİ EKLEME PROSEDÜRÜ
(
    @Ad nvarchar(50) = NULL,
    @Soyad nvarchar(50) = NULL,
    @TelNo nvarchar(15) = NULL,
    @Email nvarchar(50) = NULL,
    @AdresID int = NULL,
    @KullaniciAd nvarchar(50) = NULL,
    @Sifre nvarchar(50) = NULL,
    @KisiYetki int
)
AS
BEGIN
    INSERT INTO Kisiler(Ad, Soyad, TelNo, Email, AdresID, KullaniciAd, Sifre, KisiYetki)
    VALUES (@Ad, @Soyad, @TelNo, @Email, @AdresID, @KullaniciAd, @Sifre, @KisiYetki)
END

EXEC sp_InsertKisi
@Ad= Ali,
@Soyad= Yıldız,
@TelNo= 05354443377,
@Email= 'AliYildiz@gmail.com',
@AdresID= 12,
@KullaniciAd= Murat123,
@Sifre= '147Murat',
@KisiYetki=1
```

2) Kişi Bilgi Güncelleme Prosedürü

```
GO
CREATE PROCEDURE sp_KisiGuncelle  --2 KİŞİ BİLGİ GÜNCELLEME PROSEDÜRÜ
(
    @kisiID int,
    @Ad varchar(50),
    @Soyad varchar(50),
    @TelNo nvarchar(15),
    @Email varchar(50),
    @AdresID int = NULL,
    @KullaniciAd nvarchar(50) = NULL,
    @Sifre nvarchar(50) = NULL,
    @KisiYetki int
)
AS
BEGIN
    UPDATE Kisiler
    SET Ad=@Ad, Soyad=@Soyad, telNo=@TelNo, Email=@Email,
        AdresID=@AdresID, KullaniciAd=@KullaniciAd, Sifre=@Sifre, KisiYetki=@KisiYetki
    WHERE KisiID=@KisiID
END

/*EXEC sp_KisiGuncelle
@KisiID =1,
@Ad=Ahmet,
@Soyad= Yılmaz,
@Telno= 05443335588,
@Email= 'AhmetYilmaz@gmail.com',
@AdresID=23,
@KullaniciAd=xyz1,
@Sifre=123,
@KisiYetki=1
*/
```

3) Kişi Silme Prosedürü

```
GO
CREATE PROCEDURE sp_KisiSil  --3 KİŞİ SİLME PROSEDÜRÜ
(
    @kisiID int
)
AS
BEGIN
    DELETE FROM Kisiler WHERE KisiID=@kisiID
END

--EXEC sp_KisiSil @KisiID=1
```

4) Ada Göre Kişi Arama Prosedürü

```
GO
CREATE PROCEDURE sp_AdaGoreArama --4 ADA GÖRE KİŞİ ARAMA PROSEDÜRÜ
(
    @Ad Varchar(50)
)
AS
BEGIN
    SELECT * FROM Kisiler WHERE Ad LIKE '%' + @Ad + '%';
END

--EXEC sp_AdaGoreArama @AD= 'can'
```

5) Kayıt Tarihine Göre Rapor Listeleme

```
GO
CREATE PROCEDURE sp_TariheGoreArama --5 KAYIT TARİHİNE GÖRE RAPOR LİSTELEME
(
    @BaslangicTarih datetime,
    @BitisTarih datetime
)
AS
BEGIN
    SELECT * FROM Kisiler
    WHERE KayitTarih Between @BaslangicTarih and @BitisTarih;
END

--EXEC sp_TariheGoreArama @KayitTarih = '2023-01-01'
--veya
EXEC sp_TariheGoreArama @BaslangicTarih = '2021-12-23', @BitisTarih = '2024-10-01'
```

6) Email Adresine Göre Rapor Listeleme

```

GO
CREATE PROCEDURE sp_EmaileGoreArama --6 EMAIL ADRESİNE GÖRE RAPOR LİSTELEME
(
    @Email varchar(50)
)
AS
BEGIN
    SELECT * FROM Kisiler WHERE Email LIKE '%' + @Email + '%';
END

--EXEC sp_EmaileGoreArama @Email= 'Murat'

```

7) İçerik Türüne Göre Rapor Listeleme

```

GO
CREATE PROCEDURE sp_IcerikTureGoreArama --7 İÇERİK TÜRÜNE GÖRE RAPOR LİSTELEME
(
    @IcerikAciklama nvarchar(250)
)
AS
BEGIN
    SELECT * FROM Icerik
    WHERE IcerikAciklama LIKE 'İçecek'
    OR IcerikAciklama LIKE '%' + @IcerikAciklama + '%';
END

--EXEC sp_IcerikTureGoreArama @IcerikAciklama = 'İçecek'

```

4.4 RAPORU PDF OLARAK KAYDETME

```
using System;
using System.Data;
using Microsoft.Data.SqlClient; // Doğru kütüphane

using FastReport;
using FastReport.Export.PdfSimple; // PDF Simple Export kütüphanesi

0 basyuru
class Program
{
    0 basyuru
    static void Main()
    {
        // Bağlantı dizesi
        string connectionString = "Data Source=MONSTER\\SQLEXPRESS07;Initial Catalog=Pizzaci;Integrated Security=True;Trust Server Certificate=True;";
        string query = "SELECT IcerikAd,IcerikFiyat,IcerikAciklama FROM Icerik";

        try
        {
            // DataTable oluşturuluyor
            DataTable dataTable = new DataTable();
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                SqlDataAdapter adapter = new SqlDataAdapter(query, connection);
                adapter.Fill(dataTable); // Veriyi doldurur
            }

            // FastReport raporu oluşturuluyor
            Report report = new Report();
            report.RegisterData(dataTable, "OrderData");

            // Rapor hazırlanıyor
            report.Prepare();

            // PDF'e export ediliyor
            using (PDFSimpleExport export = new PDFSimpleExport())
            {
                export.Export(report, "OrderReport.pdf"); // PDF olarak kaydedilir
            }

            Console.WriteLine("Rapor başarıyla PDF olarak kaydedildi.");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Hata: " + ex.Message); // Hata mesajı yazdırılır
        }
    }
}
```

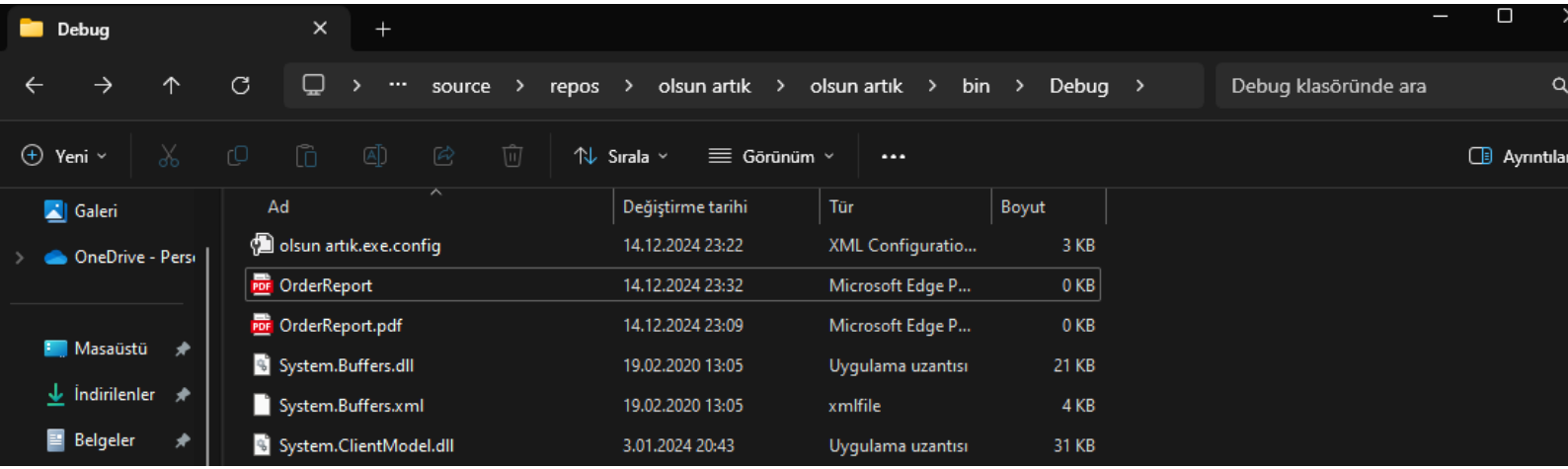
Microsoft Visual Studio Hata

Rapor başarıyla PDF olarak kaydedildi.

C:\Users\demir\source\repos\olsun artık\olsun artık\bin\Debug\olsun artık.exe (19776 işlemi), 0 koduyla çıkış yaptı (0x0).


Hata ayıklama durduğunda konsolu otomatik olarak kapatmak için Araçlar->Seçenekler->Hata Ayıklama->Hata ayıklama durduğunda konsolu otomatik olarak kapat seçeneğini etkinleştirin

Bu pencereyi kapatmak için herhangi bir tuşa basın..|



5. KULLANICI ADI VE ŞİFRE KONTROLÜ

Kullanıcı Giriş Ekranı


**OTTOMANPIZZA**

Kullanıcı Giriş Ekranı

X

Kullanıcı Adı

Şifre



Giriş Türü

☐ Yönetici ☐ Personel ☐ Müşteri

Güvenlik Kontrolü

Kayıt Ol

Temizle

Giriş Yap

0

Bu ekranda Kayıt Ol butonuna tıklayınca aşağıdaki pencere açılır.

Kişi Kayıt Ekranı


OTTOMANPIZZA


Kişi Kayıt Ekranı

X

Ad	Soyad	Telefon Numarası	E-mail Adresi
İl	İlçe	Mahalle	Sokak
Daire No	Kat	Posta Kodu	Açıklama

Temizle

 1.Adım

 2.Adım

Sonraki Adım

Burdaki bilgiler doldurulur.


OTTOMANPIZZA


Kişi Kayıt Ekranı

X

Ad	Soyad	Telefon Numarası	E-mail Adresi
ali	aslan	5051982415	aliaslan@gmail.com
İl	İlçe	Mahalle	Sokak
İstanbul	Bağcılar	fatih	1805
Daire No	Kat	Posta Kodu	Açıklama
29	2	34000	kırmızı ev

Temizle

 1.Adım

 2.Adım

Sonraki Adım

Sonraki Adım butonuna tıklayınca aşağıdaki pencereye geçiş yapılır.

OTTOMANPIZZA

Kişi Kayıt Ekranı

X

Kullanıcı Adı

ali15

Şifre

ali15

Şifre (tekrar)

ali15

Temizle

1.Adım

2.Adım

Tamamla

Tamamla butonuna tıklayınca aşağıdaki pencereye geçiş yapılır.

OTTOMANPIZZA

Kullanıcı Giriş Ekranı

X

Kullanıcı Adı

Şifre

Giriş Türü

Personel

Müşteri

Kayıt işlemi başarılı!

Kayıt Ol

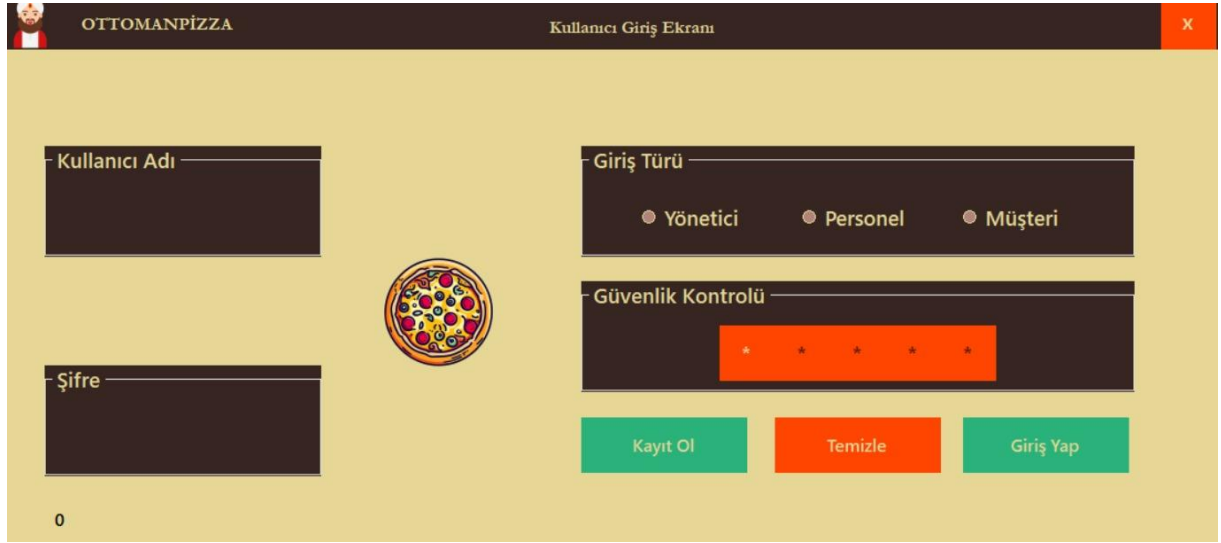
Temizle

Giriş Yap

0

Kayıt işlemi başarılı sekmesi gelir.

Aşağıdaki pencereden Müşteri kullanıcı adı ve şifresi ile güvenlik kontrolü sonrası sisteme giriş yapar.



The screenshot shows the 'Kullanıcı Giriş Ekranı' (User Login Screen) for OTTOMANPIZZA. The interface is divided into three main sections: User Information, Login Type, and Security Control. The User Information section contains two text input fields for 'Kullanıcı Adı' (Username) and 'Şifre' (Password). The Login Type section has three radio buttons: 'Yönetici' (Administrator), 'Personel' (Staff), and 'Müşteri' (Customer). The Security Control section features a password strength indicator with five stars. At the bottom, there are three buttons: 'Kayıt Ol' (Sign Up), 'Temizle' (Clear), and 'Giriş Yap' (Login). A small pizza icon is centered between the input fields. The top bar includes the OTTOMANPIZZA logo, the title 'Kullanıcı Giriş Ekranı', and a close button (X). A small '0' is visible in the bottom left corner.

6. KULLANICI TANIMLAMA VE YETKİLENDİRME

Kırmızı ok ile gösterilen yere tıklayarak sisteme yönetici olarak giriş yapma penceresine yönlendirilir.



This screenshot is identical to the one above, but with a red arrow pointing to the 'Yönetici' (Administrator) radio button in the 'Giriş Türü' (Login Type) section. This indicates the action to be taken to access the administrator login screen.

Aşağıdaki Yönlendirme Ekranı penceresi açılır.

**OTTOMANPIZZA****Yönlendirme Ekranı**

Lütfen sahip olduğunuz şifreyi personellerinizle paylaşmayınız!

Kullanıcı Adı	Şifre	Kod
<input type="text"/>	<input type="text"/>	<input type="text"/>

İşleme Devam Et

Admin bilgileri doldurulur ve İşleme Devam Et butonuna tıklanır.


**OTTOMANPIZZA****Yönlendirme Ekranı**

Lütfen sahip olduğunuz şifreyi personellerinizle paylaşmayınız!

Kullanıcı Adı	Şifre	Kod
<input type="text" value="Sthonem"/>	<input type="text" value="3444"/>	<input type="text" value="123"/>

İşleme Devam Et

Aşağıdaki Admin Panel açılır. Menü Kayıt Ekranına takılayalım:

**OTTOMANPIZZA**

Admin Panel

X

Menü Kayıt Ekranı

Çalışan Yönetim

**OTTOMANPIZZA**

Menü Oluşturma Ekranı

X

Malzeme Ekle

Malzeme

Malzeme Adı:

Fiyat:

0

Açıklama:

Mevcut Malzemeler:

	icerikID	icerikAd
▶	1	Kekik
*		

Ürün Ekle

Ürün Ekle

Ürün Adı:

Fiyat:

0

Tür:

Pizza

Boyut:

Standart

Açıklama:

Mevcut Ürünler:

	UrunID	UrunFiyat
▶	3	30,00
*		

Temizle

Oluşturmak İstediğimiz Malzeme Ürün Bilgilerini Girelim.

OTTOMANPIZZA

Menü Oluşturma Ekranı

X

Malzeme Ekle

Malzeme

Malzeme Adı:

Ketçap

Fiyat:

5

Açıklama:

Sos

Mevcut Malzemeler:

	İcerikID	İcerikAd
▶	1	Kekik
*		

Ürün Ekle

Ürün Ekle

Ürün Adı:

Sade Soda

Fiyat:

20

Tür:

İçecek

Boyut:

Standart

Açıklama:

250ml

Mevcut Ürünler:

	UrunID	UrunFiyat
▶	3	30,00
*		

Temizle

Daha Sonra Malzeme Ekle Ve Ürün Ekle butonlarına basarak içerikleri ekleyelim.

Malzeme Başarıyla Eklendi mesajı:

OTTOMANPIZZA

Menü Oluşturma Ekranı

X

Malzeme Ekle

Malzeme

Malzeme Adı:

Mayonez

Fiyat:

5

Açıklama:

Malzeme başarıyla eklendi.

Tamam

Ürün Ekle

Ürün Ekle

Ürün Adı:

Limonlu Soda

Fiyat:

20

Tür:

İçecek

Boyut:

Standart

Açıklama:

250ml

Mevcut Malzemeler:

	İçerikID	İçerikAd
▶	1	Kekik
	2	Ketçap
*		

Mevcut Ürünler:

	UrunID	UrunFiyat
▶	3	30,00
	4	20,00
*		

Temizle

Ürün Başarıyla eklendi mesajı:

OTTOMANPIZZA

Menü Oluşturma Ekranı

X

Malzeme Ekle

Malzeme

Malzeme Adı:

Mayonez

Fiyat:

5

Açıklama:

Sos

Mevcut Malzemeler:

	icerikID	icerikAd
▶	1	Kekik
	2	Ketçap
	3	Mayonez
*		

Ürün Ekle

Ürün

Ürün Adı:

Limonlu Soda

Fiyat:

20

Tür:

İçecek

Boyut:

Standart

Açıklama:

250ml

Mevcut Ürünler:

	UrunID	UrunFiyat
▶	3	30,00
	4	20,00
*		

Ürün başarıyla eklendi.

Tamam

Temizle

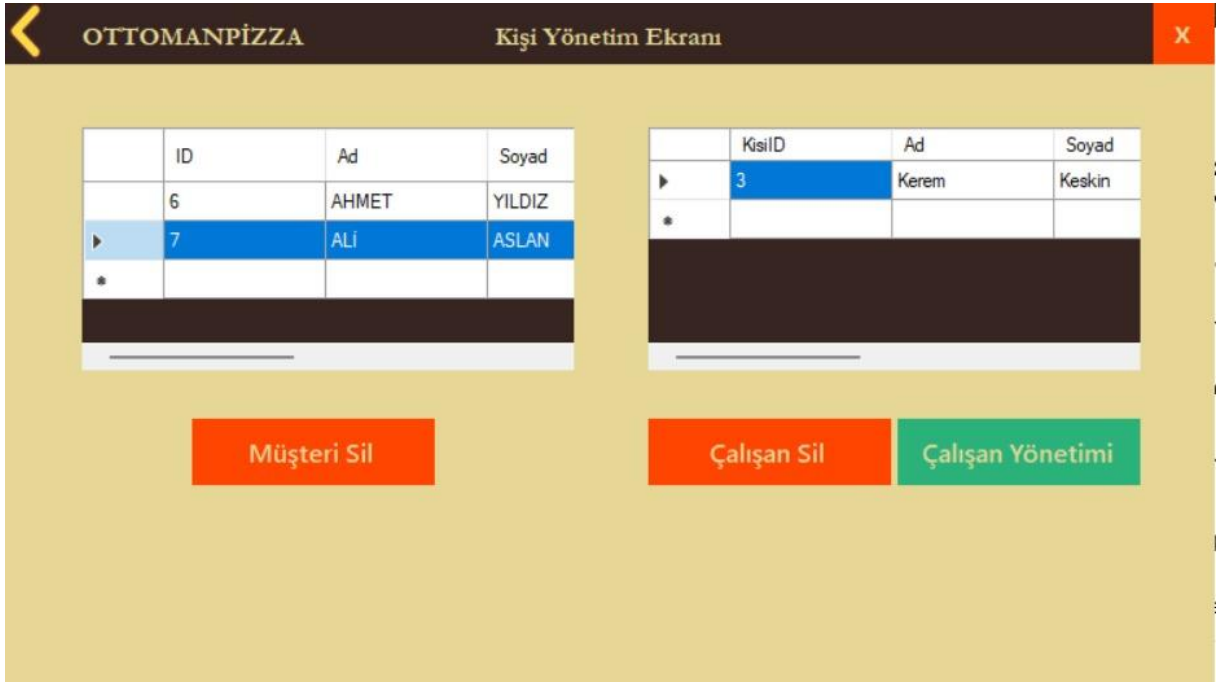
Bu sefer Admin Panel Ekranından Çalışan Yönetime girelim



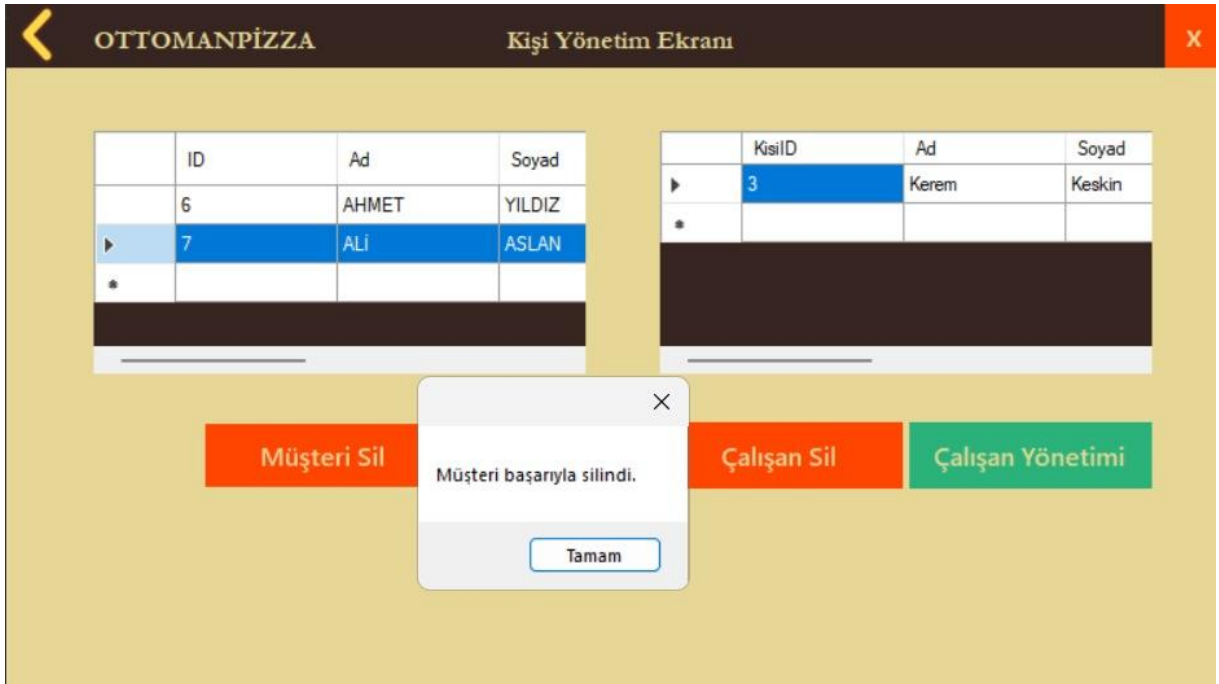
Aşağıdaki Kişi Yönetim penceresi açılır.



ID'si 7 olan Müşteriyi seçelim ve Müşteriyi Sil butonuna basalım:



“Müşteri başarıyla silindi” mesajı verdi,



ID=7 olan Müşteri silindi.

OTTOMANPIZZA

Kişi Yönetim Ekranı

X

	ID	Ad	Soyad
▶	6	AHMET	YILDIZ
*			

Müşteri Sil

	KisiID	Ad	Soyad
▶	3	Kerem	Keskin
*			

Çalışan Sil

Çalışan Yönetimi

Bu sefer Yukarıdaki ekrandan Çalışan Yönetimi butonuna tıklayalım:

OTTOMANPIZZA

Çalışan Yönetim Ekranı

X

KisiID:

İl:

Ad:

İlçe:

Soyad:

Mahalle:

TelNo:

Sokak:

Email:

Daire No: Kat:

Kullanıcı Adı:

Posta Kodu:

Şifre:

Adres Açıklama:

Çalışan Ekle

	ID	Ad
▶	3	Keren
*		

Güncelle

Yukarıda görülen sayfaya yönlendirilir.

Burada yeni bir çalışan eklemek için bilgileri dolduralım.

OTTOMANPIZZA

Çalışan Yönetim Ekranı

X

KisiID:

İl:

Manisa

Ad:

Batuhan

İlçe:

Çarşamba

Soyad:

Kıralı

Mahalle:

Gül

TelNo:

5319111125

Sokak:

Karadibi

Email:

batu15@gmail.com

Daire No:

22

Kat:

1

Kullanıcı Adı:

batu15

Posta Kodu:

34000

Şifre:

batu15

Adres Açıklama:

maniye

Çalışan Ekle

	ID	Ad
▶	3	Keren
*		

Güncelle

Çalışan Ekle Butonuna basalım.

OTTOMANPIZZA

Çalışan Yönetim Ekranı

X

KisiID:

İl:

Manisa

Ad:

Batuhan

İlçe:

Çarşamba

Soyad:

Kıralı

Mahalle:

Gül

TelNo:

5319111125

Sokak:

Karadibi

Email:

batu15@gmail.com

Daire No:

22

Kat:

1

Kullanıcı Adı:

batu15

Posta Kodu:

Şifre:

batu15

Adres Açıklama:

Çalışan Ekle

	ID	Ad
▶	3	Keren
*		

Güncelle

Yeni çalışan başarıyla eklendi.

Tamam

“Yeni çalışan başarıyla eklendi.” Mesajı aldık.

ID=3 olan Kerem Keskin Adlı çalışanın soyadını güncelleyelim

OTTOMANPIZZA

Kişi Yönetim Ekranı

X

	ID	Ad	Soyad
▶	6	AHMET	YILDIZ
*			

	KisiID	Ad	Soyad
▶	3	Kerem	Keskin
	8	Batuhan	Kıratlı
*			

Müşteri Sil

Çalışan Sil

Çalışan Yönetimi

Soyadı Yıldız olarak güncelledik.

OTTOMANPIZZA

Çalışan Yönetim Ekranı

X

KisiID: 3

İl: İstanbul

Ad: Kerem

İlçe: Bağcılar

Soyad: Keskin

Mahalle: Fatih

TelNo: 5059285731

Sokak: 1807

Email: keremdemir15@

Daire No: 2

Kat: 1

Kullanıcı Adı: kerem15

Posta Kodu: 34000

Şifre: kerem15

Adres Açıklama: kırmızı ev

Çalışan Ekle

		Soyad
▶		Yıldız
	an	Kıratlı
*		

Güncelle

Güncelle butonuna basalım.

“Çalışan bilgileri başarıyla güncellendi.” Mesajı aldık.

OTTOMANPIZZA

Çalışan Yönetim Ekranı

X

KisiID: 3

İl: İstanbul

Ad: Kerem

İlçe: Bağcılar

Soyad: Keskin

Mahalle: Fatih

TelNo: 5059285731

Sokak: 1807

Email: keremdemir15@

Daire No: 2

Kat: 1

Kullanıcı Adı: kerem15

Posta Kodu: 000

Şifre: kerem15

Adres Açıklama: nızı ev

Çalışan Ekle

		Yıldız
	an	Kıratlı
*		

Güncelle

Çalışan bilgileri başarıyla güncellendi.

Tamam

Artık çalışanımız “Kerem Yıldız” olarak güncellendi.

OTTOMANPIZZA

Çalışan Yönetim Ekranı

X

KisiID: 3

İl: İstanbul

Ad: Kerem

İlçe: Bağcılar

Soyad: Yıldız

Mahalle: Fatih

TelNo: 5059285731

Sokak: 1807

Email: keremdemir15@

Daire No: 2

Kat: 1

Kullanıcı Adı: kerem15

Posta Kodu: 34000

Şifre: kerem15

Adres Açıklama: kımızı ev

Çalışan Ekle

		Soyad
		Yıldız
	an	Kıratlı
*		

Güncelle

“Kerem Yıldız”.

		Ad	Soyad
►		Kerem	Yıldız
		Batuhan	Kıratlı
*			

Çalışan Sil

Çalışan Yönetimi

7. KAYIT SİLME VE SİLİNİMİŞ KAYITLAR TABLOSUNA EKLEME

```
-- Silme işlemini ve kaydı başka tabloya eklemeyi sağlayan prosedür
GO
CREATE PROCEDURE proc_SilKisiVeLogla
    @KisiID INT
AS
BEGIN
    -- Silinen kaydın bilgilerini SilinmisKisiler tablosuna ekleme
    INSERT INTO SilinmisKisiler (KisiID, Ad, Soyad, TelNo, Email, AdresID, SilinmeTarih)
    SELECT KisiID, Ad, Soyad, TelNo, Email, AdresID, GETDATE()
    FROM Kisiler
    WHERE KisiID = @KisiID;

    -- Kaydı Kisiler tablosundan silme
    DELETE FROM Kisiler
    WHERE KisiID = @KisiID;
END;
GO
EXEC proc_SilKisiVeLogla @KisiID = 1;
```

8. IMPORT EXPORT UYGULAMASI

9. YEDEKLE VE YEDEKTEN DÖN UYGULAMASI

YEDEKLEME UYGULAMASI

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Data.SqlClient;
6 using System.Drawing;
7 using System.Linq;
8 using System.Text;
9 using System.Threading.Tasks;
10 using System.Windows.Forms;
11
12 using System.IO;
13
14 namespace Pizzeria_Interface
15 {
16     1 bayyuru
17     public partial class FormYedekleme : Form
18     {
19         0 bayyuru
20         public FormYedekleme()
21         {
22             InitializeComponent();
23         }
24
25         1 bayyuru
26         private void InitializeComponent()
27         {
28             throw new NotImplementedException();
29         }
30
31         private string connectionString = @"Data Source=ASLAN\SQLEXPRESS;Initial Catalog=master;Integrated Security=True;TrustServerCertificate=True;";
32
33         // Yedekleme işlemi
34         0 bayyuru
35         private void btnYedekle_Click(object sender, EventArgs e)
36         {
37             SaveFileDialog saveFileDialog = new SaveFileDialog
38             {
39                 Filter = "Backup Files (*.bak)|*.bak",
40                 Title = "Veritabanı Yedeği Kaydet"
41             };
42
43             if (saveFileDialog.ShowDialog() == DialogResult.OK)
44             {
45                 string backupPath = saveFileDialog.FileName;
46
47                 try
48                 {
49                     using (SqlConnection baglanti = new SqlConnection(connectionString))
50                     {
51                         baglanti.Open();
52                         string query = $"BACKUP DATABASE PizzeriaDB TO DISK = '{backupPath}' WITH INIT, NAME = 'PizzeriaDB Backup'";
53                         SqlCommand komut = new SqlCommand(query, baglanti);
54                         komut.ExecuteNonQuery();
55                         MessageBox.Show("Veritabanı başarıyla yedeklendi!", "Başarılı", MessageBoxButtons.OK, MessageBoxIcon.Information);
56                     }
57                 }
58                 catch (Exception ex)
59                 {
60                     MessageBox.Show($"Yedekleme sırasında bir hata oluştu: {ex.Message}", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
61                 }
62             }
63         }
64     }
65 }
```

YEDEKTEN DÖN UYGULAMASI

```
// Geri Yükleme işlemi
0 baglayınız
private void btnGeriYukle_Click(object sender, EventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog
    {
        Filter = "Backup Files (*.bak)|*.bak",
        Title = "Yedekten Geri Yükle"
    };

    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        string backupPath = openFileDialog.FileName;

        try
        {
            using (SqlConnection baglanti = new SqlConnection(connectionString))
            {
                baglanti.Open();

                // Tüm aktif bağlantıları kesmek için kullanılır
                string killConnectionsQuery = @"
                DECLARE @killConnections NVARCHAR(MAX) = '';
                SELECT @killConnections += 'KILL ' + CONVERT(VARCHAR(10), spid) + '; '
                FROM master.dbo.sysprocesses
                WHERE dbid = DB_ID('PizzeriaDB');
                EXEC(@killConnections);";

                SqlCommand killCommand = new SqlCommand(killConnectionsQuery, baglanti);
                killCommand.ExecuteNonQuery();

                // Geri Yükleme
                string query = $"RESTORE DATABASE PizzeriaDB FROM DISK = '{backupPath}' WITH REPLACE, RECOVERY";
                SqlCommand komut = new SqlCommand(query, baglanti);
                komut.ExecuteNonQuery();
                MessageBox.Show("Veritabanı başarıyla geri yüklendi!", "Başarılı", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Geri yükleme sırasında bir hata oluştu: {ex.Message}", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

Uygulama ekranı – Pizza kısmı ; Pizza türü, boyutu, ekstra malzeme ve miktar seçme penceresi. Bu ekranda seçimler yapıldıktan sonra Pizza Ekle butonuna tıklanır.

Uygulama ekranı – Yan Ürün kısmı; Yan ürün türü, Miktar seçilir ve Yan Ürün Ekle butonuna basılır.

Uygulama ekranı – İçecek kısmı; İçecek türü, Miktar seçilir ve İçecek Ekle butonuna basılır.

Uygulama ana sayfasında yapılan seçimler sepette yani “Seçilen Ürünler” kısmında gözüktür. Seçilen bir ürün silinmek istenirse “Seçili Ürünü Sil” butonuna, Seçilen ürünler satın alınacaksa “Siparişi Tamamla” butonuna basılır.