

YÖNLENDİRİLMİŞ ÇALIŞMA



Araç Kiralama Otomasyonu

Emre Can
H5210013

Araç Kiralama Otomasyonu	4
Proje Ve Amaç.....	4.1
Veritabanı Tablo Veri Tipleri	5
tblUye	5.1
tblArac	5.2
tblMarka.....	6.1
tblModel.....	6.2
tblUcret	7.1
tblSegment	7.2
tblMusteri.....	8.1
tblKiradakiArac.....	8.2
Veritabanı Örnek Data Ekran Görüntüleri	9
tblArac	9.1
tblMarka	9.2
tblModel	9.3
tblSegment	9.4
tblKiradakiArac.....	9.5
tblUye	10.1
tblUcret.....	10.2
Ekran Detayları	11
E1-Kullanıcı Giriş Ekranı.....	11
E1- Görünüşü	11.1
E1-Kodları.....	11.2,12,13,14,15
E2-Kayıt Ol Ekranı	16
E2- Görünüşü	16.1
E2-Kodları.....	16.2,17,18,19,20,21,22,23
E3-Şifremi Unuttum Ekranı	24
E3- Görünüşü	24.1
E3-Kodları.....	24.2,25,26

E4-Araç Kiralama Ekranı.....	27
E4- Görünüşü	27.1
E4-Kodları.....	27.2,28,29,30,31
E5-Araç Teslim Ekranı	32
E5- Görünüşü	32.1
E5-Kodları.....	32.2,33,34,35,36
E6-Araç Yönetim Ekranı	37
E6- Görünüşü	37.1
E6-Kodları.....	37.2,38,39,40,41,42,43,44,45,46,47,48,49,50,51
E7-Müşteri Yönetim Ekranı	52
E7- Görünüşü	52.1
E7-Kodları.....	52.2,53,54,55,56,57,58,59,60,61,62
E8-Hesap Ayarları Ekranı	63
E8- Görünüşü	63.1
E8-Kodları.....	63.2,64,65,66,67,68,69
E9-Admin Ekranı	70
E9- Görünüşü	70.1
E9-Kodları.....	70.2,71,72,73,74,75,76,77,78,79

Araç Kiralama Otomasyonu

Proje ve Amaç

Araç kiralama otomasyonu projemiz, bir işletmenin işlerini kolaylaştırmak ve verileri düzenli bir şekilde depolamak için tasarlanmış bir C# form uygulamasıdır. MSSQL veritabanı kullanılarak, işletme verileri güvenli bir şekilde saklanacak ve kolayca erişilebilecektir. Aynı zamanda bu uygulama ile işletmenin faaliyetlerini daha da optimize ederek karışıklığı önlemeyi hedeflemektedir.

Uygulamanın arayüzü, kullanıcıların işletme verilerine erişmelerini sağlayacak ve verileri hızlı bir şekilde işleyebilmelerini mümkün kılacaktır. Veritabanında kaydedilen veriler, anlık raporlar oluşturmak için kullanılabilir ve bu sayede işletme yöneticileri, işlerinin genel bir görünümünü elde edebilirler.

Bunun yanı sıra, işletmenin verilerini güvenli bir şekilde saklamayı, işlemleri hızlandırmayı ve karar verme sürecinde kullanıcıların elde ettiği verilerin doğruluğunu artırmayı amaçlamaktadır.

Veritabanı Tablo Veri Tipleri

tblUye

tblUye			
	Column Name	Data Type	Allow Nulls
🔑	Id	int	<input type="checkbox"/>
	Tc	char(11)	<input type="checkbox"/>
	Ad	nvarchar(50)	<input type="checkbox"/>
	Soyad	nvarchar(50)	<input type="checkbox"/>
	TelNo	char(10)	<input type="checkbox"/>
	Mail	nvarchar(100)	<input type="checkbox"/>
	Sifre	nvarchar(20)	<input type="checkbox"/>
	YetkiDuzeyi	tinyint	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

tblArac

tblArac			
	Column Name	Data Type	Allow Nulls
🔑	Id	int	<input type="checkbox"/>
	Plaka	varchar(9)	<input type="checkbox"/>
	MarkaId	int	<input type="checkbox"/>
	ModelId	int	<input type="checkbox"/>
	Yili	char(4)	<input type="checkbox"/>
	VitesTuru	nvarchar(15)	<input type="checkbox"/>
	YakitTuru	nvarchar(20)	<input type="checkbox"/>
	SegmentId	int	<input type="checkbox"/>
	Durumu	bit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

tblMarka

tblMarka			
	Column Name	Data Type	Allow Nulls
!	Id	int	<input type="checkbox"/>
	Marka	nvarchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

tblModel

tblModel			
	Column Name	Data Type	Allow Nulls
!	Id	int	<input type="checkbox"/>
	MarkaId	int	<input type="checkbox"/>
	ModelAdi	varchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

tblUcret

tblUcret			
	Column Name	Data Type	Allow Nulls
!	Id	int	<input type="checkbox"/>
	SegmentId	int	<input type="checkbox"/>
	GunlukUcret	decimal(8, 2)	<input type="checkbox"/>
	HaftalıkUcret	decimal(8, 2)	<input type="checkbox"/>
	AylıkUcret	decimal(8, 2)	<input type="checkbox"/>
			<input type="checkbox"/>

tblSegment

tblSegment			
	Column Name	Data Type	Allow Nulls
!	Id	int	<input type="checkbox"/>
	Segment	char(1)	<input type="checkbox"/>
			<input type="checkbox"/>

tblMusteri

tblMusteri			
	Column Name	Data Type	Allow Nulls
!	Id	int	<input type="checkbox"/>
	Tc	char(11)	<input type="checkbox"/>
	EhliyetNo	nvarchar(50)	<input type="checkbox"/>
	Ad	nvarchar(50)	<input type="checkbox"/>
	Soyad	nvarchar(50)	<input type="checkbox"/>
	Mail	nvarchar(100)	<input type="checkbox"/>
	TelNo	char(10)	<input type="checkbox"/>
	Durumu	bit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

tblKiradakiArac

tblKiradakiArac			
	Column Name	Data Type	Allow Nulls
!	Id	int	<input type="checkbox"/>
	AracId	int	<input type="checkbox"/>
	MusterId	int	<input type="checkbox"/>
	KiralamaTuru	nvarchar(30)	<input type="checkbox"/>
	AlisTarihi	date	<input type="checkbox"/>
	TeslimTarihi	date	<input type="checkbox"/>
	Durumu	bit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Veritabanı Örnek Data Ekran Görüntüleri

tblArac

	Id	Plaka	MarkaId	ModelId	Yili	VitesTuru	YakitTuru	SegmentId	Durumu
1	1	34ABC34	1	1	2015	Manuel	Dizel	1	1

tblMarka

	Id	Marka
1	1	Audi

tblModel

	Id	MarkaId	ModelAdi
1	1	1	A6

tblSegment

	Id	Segment
1	1	A

tblKiradakiArac

	Id	AracId	MusteriId	KiralamaTuru	AlisTarihi	TeslimTarihi	Durumu
1	1	1	1	Haftalık	2023-03-10	2023-03-17	1

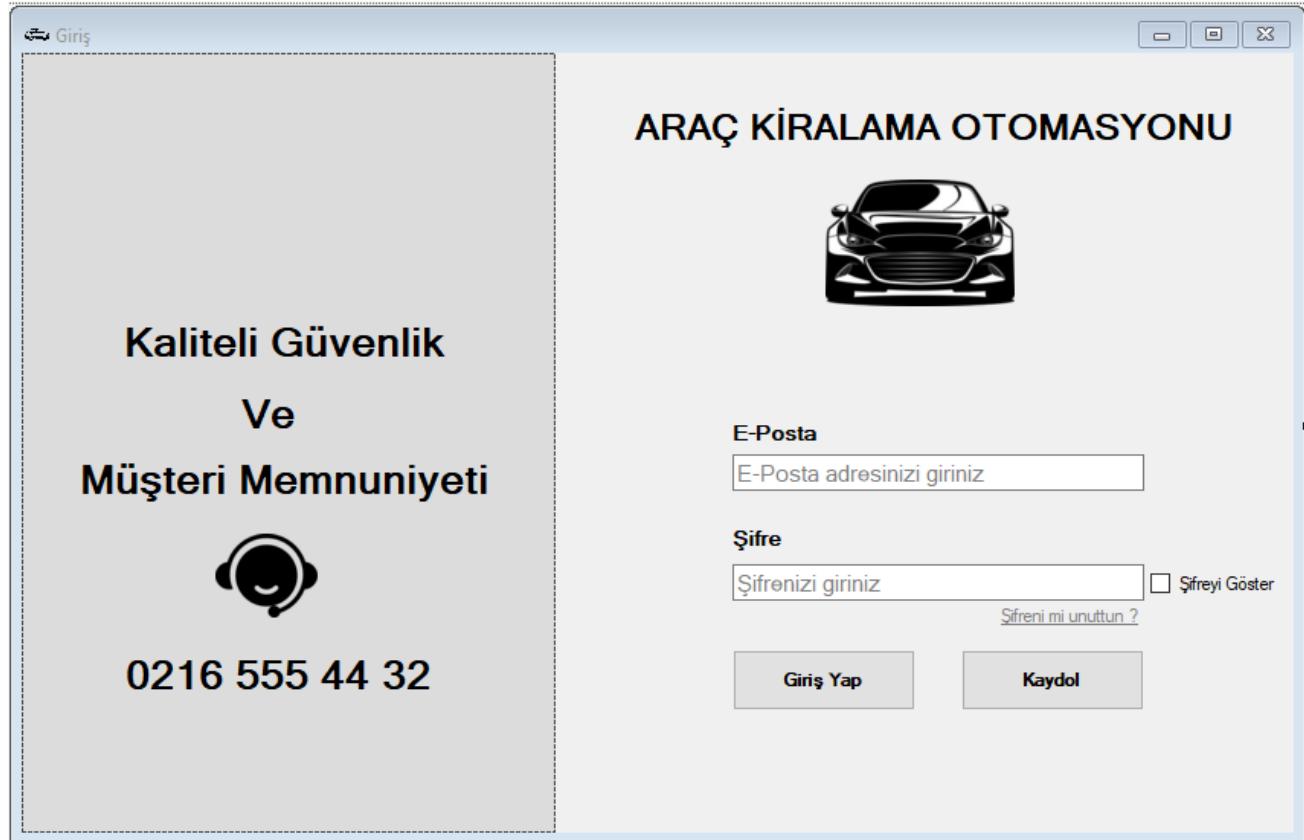
tblUye

	Id	Tc	Ad	Soyad	TelNo	Mail	Sifre	YetkiDuzeyi
1	1	52552525252	Emre	Can	5777777775	emre@gmail.com	123456	1
2	2	000000000000	admin	admin	0000000000	admin@gmail.com	admin	2
3	3	111111111111	deneme	test	5555555555	abc@gmail.com	123	0
4	13	52795215152	Emre	Can	5078976675	emrecan19o7@hotmail.com	123456	1
5	26	54577455754	beyza	koyulmuş	5555555555	beyzakoyulmus@yandex.com	Beyza123	1

tblUcret

	Id	SegmentId	GunlukUcret	HaftalıkUcret	AylıkUcret
1	1	1	500.00	3000.00	12000.00

E1-Kullanıcı Giriş Ekranı



Kodları

İlk olarak kullanıcının giriş bilgilerini veritabanında kontrol edebilmem için Sql bağlantısı oluştururdum. Daha sonra ErrorProvider nesnesi yarattım. ErrorProvider ile giriş ekranındaki textboxların kontrollerini yapıp hata var ise hata mesajı göstereceğim.

```
13
14  namespace AracKiralamaOtomasyon
15  {
16      10 basvuru
17      public partial class LoginScreen : Form
18      {
19          4 basvuru
20          public LoginScreen()
21          {
22              InitializeComponent();
23          }
24          // MSSQL Veritabanı bağlantısı kurar
25          SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-BTIS91A;Initial Catalog=Db_ArakKiralama;Integrated Security=True");
26          // Kullanıcının girdiği verilerin doğruluğunu kontrol edip hatalısa hata mesajı gösterir
27          ErrorProvider errorProvider= new ErrorProvider();
28      }
```

Burada ise “isValidEmail()” methodu ile kullanıcının girdiği emailin formata uygun olup olmadığını kontrol ediyorum.

```
28  
29 // Methods  
30 1 başvuru  
31 private bool IsValidEmail(string email) // E-postanın doğru formatta olup olmadığını kontrol eder true - false döner  
32 {  
33     if (string.IsNullOrWhiteSpace(email))  
34         return true;  
35     try  
36     {  
37         MailAddress mailAddress = new MailAddress(email); // Girilen emaili formata uygun değilse hata fırlatır ve catch bloğu çalışır  
38         return true;  
39     }  
40     catch (FormatException)  
41     {  
42         return false;  
43     }  
...}
```

E-posta ve şifre textboxlarının placeholder kodları ,

```
45 1 başvuru  
46 private void txtEposta_Enter(object sender, EventArgs e) // Textbox'a girildiğine placeholder silmek için  
47 {  
48     if (txtEposta.Text == "E-Posta adresinizi giriniz")  
49     {  
50         txtEposta.Text = "";  
51         txtEposta.ForeColor = Color.Black;  
52     }  
53  
54 1 başvuru  
55 private void txtEposta_Leave(object sender, EventArgs e) // Textboxdan çıktıduğunda eğer veri girilmediyse placeholderı tekrar yazdırır  
56 {  
57     if (txtEposta.Text == "")  
58     {  
59         txtEposta.Text = "E-Posta adresinizi giriniz";  
60         txtEposta.ForeColor = Color.Gray;  
61     }  
62  
63  
64 1 başvuru  
65 private void txtPassword_Leave(object sender, EventArgs e) // Textboxdan çıktıduğunda eğer veri girilmediyse placeholderı tekrar yazdırır  
66 {  
67     if (txtPassword.Text == "")  
68     {  
69         txtPassword.PasswordChar = '\u0000'; // şifreyi görünür yap  
70         txtPassword.Text = "Şifrenizi giriniz";  
71         txtPassword.ForeColor = Color.Gray;  
72     }  
73  
74 1 başvuru  
75 private void txtPassword_Enter(object sender, EventArgs e) // Textbox'a girildiğine placeholder siler  
76 {  
77     if (txtPassword.Text == "Şifrenizi giriniz")  
78     {  
79         txtPassword.Text = "";  
80         if (!checkBoxShowPassword.Checked) // şifreyi göster checkboxı işaretli değilse  
81             txtPassword.PasswordChar = '*'; // şifreyi sansürle  
82         txtPassword.ForeColor = Color.Black;  
83     }  
84 }
```

Textboxa focusluyken boş bir yere tıklanma ve açılışta textboxa olan odağı kaldırmak için gerekli olan eventler,

```
--  
86 // Başlangıçta textboxa focus olmasını öner  
87 1 başvuru  
88 private void LoginScreen_Shown(object sender, EventArgs e)  
89 {  
90     lblEposta.Focus();  
91 }  
92  
93 // Ekranda boş bir yere tıklandığında textbox odağını kaldırmak için  
94 1 başvuru  
95 private void LoginScreen_Click(object sender, EventArgs e)  
96 {  
97     lblEposta.Focus();  
98 }  
99  
100 1 başvuru  
101 private void panel1_Click(object sender, EventArgs e)  
102 {  
103     lblEposta.Focus();  
104 }
```

Kayıt ol , Şifremi unuttum ekranlarının gösterilmesi ve giriş ekranının kapatılması durumunda uygulamanın sonlandırılması işlemi için gereken kod blokları :

```
104 // Kayıt olma formunu açar ve bu formu gizler
105 1 başvuru
106 private void btnKaydol_Click(object sender, EventArgs e)
107 {
108     RegisterScreen registerScreen= new RegisterScreen();
109     registerScreen.Show();
110     this.Hide();
111 }
112
113 // ekran kapatılırsa çalışacak kod bloğu
114 1 başvuru
115 private void LoginScreen_FormClosing(object sender, FormClosingEventArgs e)
116 {
117     Application.Exit(); // uygulamayı sonlandır
118 }
119
120 // Şifre sıfırlama ekranını açar
121 1 başvuru
122 private void lblSifreReset_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
123 {
124     ResetPasswordScreen resetPasswordScreen = new ResetPasswordScreen();
125     resetPasswordScreen.Show();
126 }
```

Eposta textboxının uygun formatta olup olmama durumlarının belirlenmesi , hata mesajı gösterilmesi için gerekli olan kod satırları :

```
201 1 başvuru
202 private void txtEposta_Validating(object sender, CancelEventArgs e) //textbox içerişine girilen verinin formatını kontrol eder
203 {
204     string email = txtEposta.Text;
205     bool isValid = IsValidEmail(email); // Girilen değer eposta mı ?
206     if (!isValid) // Eposta değilse
207     {
208         // Hata mesajının görüntülenme durumları
209         if (txtEposta.Text == "E-Posta adresinizi giriniz")
210         {
211             txtEposta.BackColor = Color.White;
212             errorProvider.SetError(txtEposta, "");
213         }
214         else
215         {
216             // Eposta yanlış formatta
217             txtEposta.BackColor = Color.FromArgb(255, 114, 86); // arkaplanı kırmızı yapar
218             txtEposta.ForeColor = Color.Black; // yazı rengini siyah yapar
219             errorProvider.SetError(txtEposta, "Lütfen geçerli bir eposta adresi girin."); //Hata mesajı gösterir
220             e.Cancel = false; // Hata varken textboxdan çıkmaya izin verir
221         }
222     }
223     else
224     {
225         // Eposta doğru formatlidir
226         txtEposta.BackColor = Color.White; // arkaplanı beyaz yapar
227         txtEposta.ForeColor = Color.Black; // yazı rengini siyah yapar
228         errorProvider.SetError(txtEposta, ""); // hata mesajını kaldırır
229         e.Cancel = false;
230     }
231 }
232 }
```

Giriş yap butonuna tıklandığında girilen eposta ve şifrenin veritabanındaki kayıtlardan biriyle eşleşmesi sonucu yetki düzeyine göre kullanıcının ilgili ekranaya yönlendirilmesi ve eşleşen kullanıcı bilgisi olmaması durumunda hata mesajı yazdırılması.

```
1 basvuru
127 private void btnGirisYap_Click(object sender, EventArgs e)
128 {
129     if (txtEposta.Text == "E-Posta adresinizi giriniz" || txtPassword.Text == "Şifrenizi Giriniz") // Eposta - şifre boş ise
130     {
131         MessageBox.Show("Lütfen E-Posta ve şifrenizi eksiksiz girin.");
132     }
133     else
134     {
135
136         try
137         {
138             baglanti.Open(); // veritabanı bağlantısını aktif et
139
140             //sql sorgusu oluşturma
141             SqlCommand cmd = new SqlCommand("select Mail , Sifre , YetkiDuzeyi , Ad from tblUye where Mail=@mail and Sifre=@sifre",baglanti);
142
143             //Parametre geçiriliyor...
144             cmd.Parameters.AddWithValue("@mail", txtEposta.Text);
145             cmd.Parameters.AddWithValue("@sifre", txtPassword.Text);
146             // Data okuyucu
147             SqlDataReader reader;
148             reader = cmd.ExecuteReader(); // Sorguyu execute et过分的注释导致了代码的冗余，建议精简注释。
149             if(reader.Read()) // Girilen eposta ve şifreyle eşleşen kayıt varsa
150             {
151                 string permission = reader.GetValue(2).ToString(); // Sorğu sonucundan YetkiDuzeyi (2 nolu index) çekip permission değişkenine at
152                 string name = reader.GetValue(3).ToString(); // 3 nolu index (Ad) verisini name değişkenine at
153
154                 switch(permission) // Giriş yapan üyenin türünü kontrol et
155                 {
156                     case "1": // permission 1 ise normal üye
157                     {
158                         MainScreen mainScreen = new MainScreen();
159                         mainScreen.ad = name; // Ana ekranı kullanıcının adını aktarma
160                         mainScreen.Show(); // Ana ekranı göster
161                         this.Hide(); // Bu ekranı gizle
162                         break;
163                     }
164                     case "2": // permission 2 ise admin
165                     {
166                         MessageBox.Show("Admin giriş'i");
167                         break;
168                     }
169                     default: // aksi durumda pasif üye
170                     {
171                         MessageBox.Show("Üyeliğiniz pasif durumda lütfen yöneticiniz ile iletişime geçin.");
172                         break;
173                     }
174                 }
175             }
176             else // Girilen Eposta ve şifreyle eşleşen kayıt yoksa
177             {
178                 if (txtPassword.Text == "Şifrenizi giriniz")
179                 {
180                     MessageBox.Show("Lütfen E-Posta ve şifrenizi eksiksiz girin.");
181                 }
182                 else
183                     MessageBox.Show("E-Posta veya Şifre Hatalı !");
184
185             }
186
187         }
188         catch (Exception ex)
189         {
190             MessageBox.Show(ex.ToString()); // try bloğu içerisinde herhangi bir hata olursa hatayı yazdır.
191         }
192     }
193     finally
194     {
195         baglanti.Close(); // Veritabanı bağlantısını kapatır
196     }
197 }
198 }
```

Şifreyi göster checkboxının işaretlenip işaretlenmemesi durumunda çalışacak kod blogu.

```
234 // Login ekranı şifreyi göster checkboxı durum değişimi
235 1 başvuru
236 private void checkBoxShowPassword_CheckedChanged(object sender, EventArgs e)
237 {
238     if(checkBoxShowPassword.Checked)    // checkbox işaretliyse
239     {
240         txtPassword.PasswordChar = '\u0000';    // şifreyi göster
241     }
242     else
243     {
244         // İşaretli değilse şifreyi gösterme
245         // textboxın boş olup olmama durumu ve placeholder yazılı olma durumunun kontrolü
246         if (!String.IsNullOrEmpty(txtPassword.Text) && txtPassword.Text != "Şifrenizi giriniz")
247             txtPassword.PasswordChar = '*';
248     }
249 }
250 }
```

E2- Kayıt Ol Ekranı

KAYIT OL

TC: TC Kimlik No

Ad: Adınız

Soyad: Soyadınız

Telefon No: Telefon Numaranız

E-Posta: E-Posta

Şifre: Şifre

Şifre Tekrar: Şifre Tekrar

Şifreyi Göster

**Kaliteli Güvenlik
Ve
Müşteri Memnuniyeti**

0216 555 44 32

Kodlar

Veritabanı bağlantısı , ErrorProvider nesnesinin oluşturulması işlemleri

```
18  namespace ArackiralamaOtomasyon
19  {
20      5 basvuru
21      public partial class RegisterScreen : Form
22      {
23          1 basvuru
24          public RegisterScreen()
25          {
26              InitializeComponent();
27          }
28          // veritabanı connection işlemi
29          SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-BTIS91A;Initial Catalog=Db_Arackiralama;Integrated Security=True");
30
31          // error provider nesnesi oluşturuluyor
32          ErrorProvider errorProvider = new ErrorProvider();
```

Kayıt ol ekranı açıldığında textboxdan focusu alma, geri butonunun gerçekleştireceği işlem, form kapatıldığında çalışacak kod satırları :

```
33 // form kapatıldığında yapılacak işlemler
34 1 başvuru
35 private void RegisterScreen_FormClosing(object sender, FormClosingEventArgs e)
36 {
37     LoginScreen loginScreen = new LoginScreen(); // login ekranını oluştur
38     loginScreen.Show(); // login ekranını göster
39 }
40
41 // geri butonuna tıklandığında yapılacak işlemler
42 1 başvuru
43 private void btnBack_Click(object sender, EventArgs e)
44 {
45     this.Close(); // bu ekranı kapat
46 }
47
48 1 başvuru
49 private void RegisterScreen_Shown(object sender, EventArgs e)
{
    lblEposta.Focus(); // kayıt ol ekranı ilk açılışta focusu textboxdan almak için labela focuslan
}
```

Kayıt ol ekranında girilen telefon numarası , eposta değerlerinin formatlarının kontrolü için gereken metodların oluşturulması:

```
138 1 başvuru
139 private bool IsValidPhoneNumber(string phoneNumber) // Verilen parametre telefon numarası mı kontrol ediliyor
140 {
141     bool isValid = true;
142
143     // Telefon numarasının uzunluğunu kontrol ediliyor.
144     if (phoneNumber.Length != 10)
145     {
146         isValid = false;
147     }
148
149     // Telefon numarasının yalnızca sayılardan olduğu kontrol ediliyor.
150     foreach (char c in phoneNumber)
151     {
152         if (!char.IsDigit(c))
153         {
154             isValid = false;
155             break;
156         }
157     }
158
159     return isValid;
160 }
161
162 1 başvuru
163 private bool IsValidEmail(string email) // Verilen parametre email formatına uygunluğu kontrol ediliyor
164 {
165     if (string.IsNullOrWhiteSpace(email)) // Eğer email boş ya da null ise, true döner
166     {
167         return true;
168     }
169     try // Eğer email geçerli bir e-posta formatında değilse, false döndürülür. Aksi takdirde true döndürülür.
170     {
171         MailAddress mailAddress = new MailAddress(email);
172         return true;
173     }
174     catch (FormatException)
175     {
176         return false;
177     }
178 }
```

Kaydol butonuna tıklandığında kullanıcının girdiği bilgilerin veritabanında bulunamaması halinde kullanıcıyı veritabanına insert edecek kod satırları :

```
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
```

```
private void btnKaydol_Click(object sender, EventArgs e) // kaydol butonuna tıklandığına çalışacak kodlar
{
    // Tüm hataların düzeltilmesi kontrol ediliyor.
    if (
        errorProvider.GetError(txtTcKimlik) == ""
        && errorProvider.GetError(txtAd) == ""
        && errorProvider.GetError(txtSoyad) == ""
        && errorProvider.GetError(txtNumara) == ""
        && errorProvider.GetError(txtEposta) == ""
        && errorProvider.GetError(txtPassword) == ""
        && errorProvider.GetError(txtPasswordRepeat) == "")
    {
        // Textboxların içerisindeki herhangi bir veri girişi olmadığından
        if(
            txtTcKimlik.Text=="TC Kimlik No"
            || txtAd.Text == "Adınız"
            || txtSoyad.Text == "Soyadınız"
            || txtNumara.Text == "Telefon Numaranız"
            || txtEposta.Text=="E-Posta"
            || txtPassword.Text == "Şifre"
            || txtPasswordRepeat.Text=="Şifre Tekrar")
        {
            MessageBox.Show("Lütfen tüm bilgileri eksiksiz doldurun.");
        }
        // Textboxlar boş değilse
        else
        {
            try
            {
                baglanti.Open();           // veritabanı bağlantısını aç
                SqlCommand ControlCmd = new SqlCommand("select Tc,Mail from tblUye where Tc = @tc or Mail=@mail ", baglanti);
                // parametre geçiriliyor...
                ControlCmd.Parameters.AddWithValue("@tc", txtTcKimlik.Text);
                ControlCmd.Parameters.AddWithValue("@mail", txtEposta.Text);
                SqlDataReader reader; // data okuyucu tanımla

                reader = ControlCmd.ExecuteReader(); // Sorguyu execute et ve dönen sonucu reader değişkenine at
                if (reader.Read()) // Girilen bilgilerle eşleşen kayıt bulunursa
                {
                    MessageBox.Show("Eşleşen kayıt bulundu !");
                }
                else
                {
                    reader.Close(); // data okuyucuyu kapat
                    // Üye bilgilerini tblUye tablosuna insert et
                    SqlCommand cmd =
                        new SqlCommand("INSERT INTO tblUye (Tc, Ad,Soyad,TelNo,Mail,Sifre) " +
                                      "VALUES (@value1, @value2,@value3,@value4,@value5,@value6)", baglanti);

                    //parametre geçiriliyor...
                    cmd.Parameters.AddWithValue("@value1", txtTcKimlik.Text);
                    cmd.Parameters.AddWithValue("@value2", txtAd.Text);
                    cmd.Parameters.AddWithValue("@value3", txtSoyad.Text);
                    cmd.Parameters.AddWithValue("@value4", txtNumara.Text);
                    cmd.Parameters.AddWithValue("@value5", txtEposta.Text);
                    cmd.Parameters.AddWithValue("@value6", txtPassword.Text);
                    cmd.ExecuteNonQuery(); // sorguyu execute et

                    MessageBox.Show("Kayıt başarıyla gerçekleşti !");

                    this.Close(); // bu formu kapat
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Hata oluştu : " + ex.Message);
            }
            finally
            {

                baglanti.Close();      // veritabanı bağlantısını
            }
        }
    }
    else //Error provider aktif olan textbox varsa kayıt işlemi gerçekleşmez
    {
        MessageBox.Show("Lütfen hataları düzeltin.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Textboxların placeholder kodları :

```
1 başvuru
178     private void txtTcKimlik_Enter(object sender, EventArgs e) //textboxa girildiğinde yapılacak placeholder işlemi
179     {
180         if (txtTcKimlik.Text == "TC Kimlik No")           //textbox da placeholder duruyor ise
181         {
182             txtTcKimlik.Text = "";                         //textbox içerisindeki placeholderı temizle
183             txtTcKimlik.ForeColor = Color.Black;          // yazı rengini siyah yap
184         }
185     }
186
187
188     1 başvuru
189     private void txtTcKimlik_Leave(object sender, EventArgs e)
190     {
191         if (txtTcKimlik.Text == "") // textbox boş ise
192         {
193             txtTcKimlik.Text = "TC Kimlik No"; // placeholder ekle
194             txtTcKimlik.ForeColor = Color.Gray; // yazı rengini gri yap
195         }
196     }
197
198     1 başvuru
202     private void txtAd_Leave(object sender, EventArgs e)
203     {
204         if (txtAd.Text == "") //textbox boş ise
205         {
206             txtAd.Text = "Adınız"; // textbox içerisinde placeholder ekle
207             txtAd.ForeColor = Color.Gray; // yazı rengini gri yap
208         }
209     }
210
211     1 başvuru
211     private void txtAd_Enter(object sender, EventArgs e)
212     {
213         if (txtAd.Text == "Adınız") // textbox içerisinde placeholder var ise
214         {
215             txtAd.Text = ""; //textbox içerisindeki placeholderı temizle
216
217             txtAd.ForeColor = Color.Black; // yazı rengini siyah yap
218         }
219     }
220
221     1 başvuru
227     private void txtSoyad_Leave(object sender, EventArgs e)
228     {
229         if (txtSoyad.Text == "") // textbox boş ise
230         {
231             txtSoyad.Text = "Soyadınız"; // placeholder ekle
232             txtSoyad.ForeColor = Color.Gray; // yazı rengini gri yap
233         }
234     }
235
236     1 başvuru
236     private void txtSoyad_Enter(object sender, EventArgs e)
237     {
238         if (txtSoyad.Text == "Soyadınız") // placeholder var ise
239         {
240             txtSoyad.Text = ""; // placeholder sil
241
242             txtSoyad.ForeColor = Color.Black; // yazı rengini siyah yap
243         }
244     }
```

```

1 başvuru
257     private void txtNumara_Leave(object sender, EventArgs e)
258     {
259         if (txtNumara.Text == "") // textbox boş ise
260         {
261             txtNumara.Text = "Telefon Numaranız"; // placeholder ekle
262             txtNumara.ForeColor = Color.Gray; // yazı rengini gri yap
263         }
264     }
265
266
1 başvuru
267     private void txtNumara_Enter(object sender, EventArgs e)
268     {
269         if (txtNumara.Text == "Telefon Numaranız") // placeholder var ise
270         {
271             txtNumara.Text = ""; // placeholder sil
272
273             txtNumara.ForeColor = Color.Black; // yazı rengini siyah yap
274         }
275     }
1 başvuru
277     private void txtEposta_Enter(object sender, EventArgs e)
278     {
279         if (txtEposta.Text == "E-Posta") // placeholder var ise
280         {
281             txtEposta.Text = ""; // placeholder sil
282
283             txtEposta.ForeColor = Color.Black; // yazı rengini siyah yap
284         }
285     }
286
1 başvuru
287     private void txtEposta_Leave(object sender, EventArgs e)
288     {
289         if (txtEposta.Text == "") // textbox boş ise
290         {
291             txtEposta.Text = "E-Posta"; // placeholder ekle
292             txtEposta.BackColor = Color.White; // arkaplan rengini beyaz yap
293             txtEposta.ForeColor = Color.Gray; // yazu rengini gri yap
294         }
295
1 başvuru
311     private void txtPassword_Enter(object sender, EventArgs e)
312     {
313         if (txtPassword.Text == "Şifre") // placeholder var ise
314         {
315             txtPassword.Text = ""; // placeholder sil
316             if (!checkBoxShowPassword.Checked) // şifreyi göster checkbox işaretli değilse
317                 txtPassword.PasswordChar= '*'; // şifreyi yıldızla
318             txtPassword.ForeColor = Color.Black; // yazı rengini siyah yap
319         }
320     }

```

```

1 basvuru
300  private void txtPassword_Leave(object sender, EventArgs e)
301  {
302      if (txtPassword.Text == "") // textbox boş ise
303      {
304          txtPassword.PasswordChar = '\u0000'; // şifre alanını görünürl yap
305          txtPassword.Text = "Şifre"; // placeholder ekle
306          txtPassword.ForeColor = Color.Gray; // yazı rengini gri yap
307      }
308  }
309
310
311
312  1 basvuru
313  private void txtPasswordRepeat_Leave(object sender, EventArgs e) // textbox bırakıldığında yapılacak işlemler
314  {
315      if (txtPasswordRepeat.Text == "") // textbox boş ise
316      {
317          txtPasswordRepeat.PasswordChar = '\u0000'; // şifreyi sansürleme
318          txtPasswordRepeat.Text = "Şifre Tekrar"; // placeholder yaz
319          txtPasswordRepeat.ForeColor = Color.Gray; // yazı rengini gri yap
320      }
321
322
323
324  1 basvuru
325  private void txtPasswordRepeat_Enter(object sender, EventArgs e)
326  {
327      if (txtPasswordRepeat.Text == "Şifre Tekrar") // placeholder var ise
328      {
329          txtPasswordRepeat.Text = ""; // placeholder sil
330          if (!checkBoxShowPassword.Checked) // şifreyi göster işaretli değilse
331              txtPasswordRepeat.PasswordChar = '*'; // şifreyi yıldızla
332
333          txtPasswordRepeat.ForeColor = Color.Black; // yazı rengini siyah yap
334      }
335  }
336
337
338
339
340
341
342

```

Sadece sayı veya harf girilmesi için gereken kod satırları :

```

1 basvuru
197  private void txtTcKimlik_KeyPress(object sender, KeyPressEventArgs e)
198  {
199      e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar); // Yalnızca sayısal değerlerin girilmesine izin verir.
200  }
201
202
203
204
205
206
207
208
209
210
211  1 basvuru
212  private void txtAd_KeyPress(object sender, KeyPressEventArgs e)
213  {
214      e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
215          && !char.IsSeparator(e.KeyChar); // sadece harf girişi
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246  1 basvuru
247  private void txtSoyad_KeyPress(object sender, KeyPressEventArgs e)
248  {
249      e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
250          && !char.IsSeparator(e.KeyChar); // sadece harf girişi
251
252
253
254
255
256

```

Textboxlara girilen değerin hatalı olup olmama durumunun kontrolü için gereken kod blokları :

```
1 başvuru
354     private void txtEposta_Validate(object sender, CancelEventArgs e)
355     {
356         string email = txtEposta.Text; // textbox içerisindeki veriyi email değişkenine at
357         bool isValid = IsValidEmail(email); // email formatına uygun olup olmadığını kontrol ediyor
358         if (!isValid) // email doğru formatta değilse
359         {
360             if (txtEposta.Text == "E-Posta") //placeholder var ise
361             {
362                 txtEposta.BackColor = Color.White; // arkaplan rengini beyaz yap
363                 errorProvider.SetError(txtEposta, ""); // hata mesajını kaldır
364             }
365             else
366             {
367                 // Eposta yanlış formatta
368                 txtEposta.BackColor = Color.FromArgb(255, 114, 86); // arka plan rengini kırmızı yap
369                 txtEposta.ForeColor = Color.Black; // yazı rengini siyah yap
370                 errorProvider.SetError(txtEposta, "Lütfen geçerli bir eposta adresi girin."); // hata mesajı görüntüle
371                 e.Cancel = false; // hata varken textboxdan ayrılmama durumunu kaldır
372             }
373         }
374     }
375     else
376     {
377         // Eposta doğru formatlıdır
378         txtEposta.BackColor = Color.White;
379         txtEposta.ForeColor = Color.Black;
380         errorProvider.SetError(txtEposta, "");
381         e.Cancel = false;
382     }
383
384 }
385
386
387     1 başvuru
388     private void txtNumara_Validate(object sender, CancelEventArgs e) // telefon numarası doğrulama
389     {
390         // e.Cancel = false;
391         string phoneNumber = txtNumara.Text;
392         if (!IsValidPhoneNumber(phoneNumber)) // girilen değer telefon numarası değilse
393         {
394             if (txtNumara.Text == "Telefon Numaranız") // placeholder yazılı ise
395             {
396                 txtNumara.BackColor = Color.White; // arkaplan rengini beyaz yap
397                 errorProvider.SetError(txtNumara, ""); // hata mesajını kaldır
398             }
399             else
400             {
401                 // Eposta yanlış formatta
402                 txtNumara.Select(0, txtNumara.Text.Length); // numarayı seçili hale getir (silmeyi kolaylaştırmak için)
403                 txtNumara.BackColor = Color.FromArgb(255, 114, 86); // arkaplan rengini kırmızı yap
404                 txtNumara.ForeColor = Color.Black; // yazı rengini siyah yap
405                 errorProvider.SetError(txtNumara, "Lütfen geçerli bir telefon numarası girin.Örnek: 5554443322"); // hata mesajı göster
406             }
407         }
408     }
409     else
410     {
411         txtNumara.BackColor = Color.White; // arkaplan rengini beyaz yap
412         txtNumara.ForeColor = Color.Black; // yazı rengini siyah yap
413         errorProvider.SetError(txtNumara, ""); // hata mesajını kaldır
414     }
415
416     1 başvuru
417     private void txtPassword_Validate(object sender, CancelEventArgs e) // doğrulama eventi
418     {
419         string sifre = txtPassword.Text.Trim(); // şifrede boşluk varsa siler
420         // Şifre en az 8 karakter uzunluğunda ve en az bir büyük harf, bir küçük harf ve bir rakam içermelidir.
421         string pattern = @"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[a-zA-Z\d]{8}$";
422
423         if (!Regex.IsMatch(sifre, pattern)) // şifre kurallara uygun değil ise
424         {
425             // hata mesajı göster
426             errorProvider.SetError(txtPassword, "Şifre en az 8 karakter uzunluğunda olmalı ve en az bir büyük harf, bir küçük harf ve bir rakam içermelidir.");
427         }
428     }
429     else // şifre kurallara uygunsa
430     {
431         errorProvider.SetError(txtPassword, ""); // hata mesajını kaldır
432     }
433 }
434
```

```

435     1 başvuru
436     private void txtPasswordRepeat_Validating(object sender, CancelEventArgs e) // şifre doğrulama eventi
437     {
438         string sifreTekrar = txtPasswordRepeat.Text.Trim(); // şifre tekrarında boşluk varsa bunu kaldırıp sifreTekrar değişkenine atar
439         if (sifreTekrar != txtPassword.Text.Trim()) // şifre tekrarı , şifre ile aynı değilse
440         {
441             errorProvider.SetError(txtPasswordRepeat, "Şifre tekrarı, şifre ile aynı olmalıdır."); // hata mesajı görüntüle
442         }
443         else
444         {
445             errorProvider.SetError(txtPasswordRepeat, ""); // hata mesajını kaldır
446         }
447     }

```

Şifreyi göster checkboxının işaretlenme durumu :

```

449     1 başvuru
450     private void checkBoxShowPassword_CheckedChanged(object sender, EventArgs e) // şifreyi göster checkbox durum değişimi
451     {
452         if (checkBoxShowPassword.Checked) // checkbox işaretliyse
453         {
454             txtPassword.PasswordChar = '\0'; // şifreyi göster
455             txtPasswordRepeat.PasswordChar = '\0'; // şifre tekrarını göster
456         }
457         else
458         {
459             if (!String.IsNullOrEmpty(txtPasswordRepeat.Text) && txtPasswordRepeat.Text != "Şifre Tekrar") // şifre tekrarı boş değilse ve placeholder yok ise
460                 txtPasswordRepeat.PasswordChar = '*'; // şifre tekrarını sansürle
461
462             if (!String.IsNullOrEmpty(txtPassword.Text) && txtPassword.Text != "Şifre") // şifre boş veya null değilse ve placeholder yok ise
463                 txtPassword.PasswordChar = '*'; // şifreyi sansürle
464         }
465     }
466 }
467

```

E3-Şifremi Unuttum Ekranı



Kodlar

Öncelikle diğer ekranlarda olduğu gibi veritabanı bağlantısı ve ErrorProvider nesnesini tanımlıyorum.

```
14  namespace AracKiralamaOtomasyon
15  {
16      4 basıru
17      public partial class ResetPasswordScreen : Form
18      {
19          1 basıru
20          public ResetPasswordScreen()
21          {
22              InitializeComponent();
23          }
24          // veritabanı bağlantısı oluştur
25          SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-BTIS91A;Initial Catalog=Db_AracKiralama;Integrated Security=True");
26          //error provider nesnesini yarat
27          ErrorProvider errorProvider= new ErrorProvider();
```

Şifre sıfırlamak için mail gönderimi yapmama yardımcı olacak olan kütüphaneyi projeme import ediyorum.

```
7  || using System.Net.Mail; // Mail Gönderimi için
```

Mail gönderimi için bir method oluşturuyorum. Bu method kullanıcının parametre olarak传递的 mail adresine mail gönderimi yapacak.

```
30  private void SifreGonder(string aliciEposta)
31  {
32      string ad = "";
33      string passFromDb = "";
34
35      try
36      {
37          baglanti.Open(); // veritabanı bağlantısını aç
38          SqlCommand cmd = new SqlCommand("select Ad,Mail , Sifre from tblUye where Mail=@mail", baglanti);
39          cmd.Parameters.AddWithValue("@mail", txtEposta.Text); // @mail parametresini geç
40          SqlDataReader reader; // data okuyucu tanımlama
41          reader = cmd.ExecuteReader(); // soruyu execute et ve sonucu reader'a aktar
42          if (reader.Read()) // sonuc var ise
43          {
44              ad = reader.GetString(0);
45              passFromDb = reader.GetString(2);
46
47
48              MailMessage mail = new MailMessage(); // mesaj gönderme kütüphanesi
49              SmtpClient smtpServer = new SmtpClient("smtp.gmail.com"); // gmail smtp adresi
50
51              mail.From = new MailAddress("arackiralamayc@gmail.com"); // maili gönderecek olan eposta
52              mail.To.Add(aliciEposta); // alici eposta adresi
53              mail.Subject = "Araç Kiralama Otomasyonu Şifreniz"; // mail konusu
54              mail.Body = "Araç Kiralama Otomasyonu Giriş Şifreniz: " + passFromDb; // mail içeriğine db den çekilen passwordu yazdır.
55
56              smtpServer.Port = 587; // gmail smtp portu
57              smtpServer.Credentials = new NetworkCredential("arackiralamayc@gmail.com", "rniskylvicjdzcvj"); // mail gönderecek hesabın eposta - şifresi
58              smtpServer.EnableSsl = true; //SSL aktif et
59
60              smtpServer.Send(mail); // maili gönder
61              MessageBox.Show("Şifreniz " + aliciEposta + " adresine gönderildi."); // kullanıcıya mesajın gönderildiğine dair mesaj ver
62              this.Close(); // ekranı kapat
63
64      }
65      else // girilen eposta ile eşleşen kayıt bulunamazsa
66      {
67          MessageBox.Show("Girdiğiniz E-Posta ile eşleşen kayıt bulunamadı !");
68          txtEposta.Text = "E-Posta adresinizi giriniz";
69          txtEposta.ForeColor = Color.Gray;
70      }
71  }
72
73  catch (Exception ex) // try bloğu içerisinde bir hata oluştuğunda
74  {
75      MessageBox.Show(ex.ToString());
76  }
77  finally
78  {
79      baglanti.Close(); // veritabanı bağlantısını kapat
80  }
```

Girilen mail adresinin doğru formatta olup olmadığını kontrol etmek için “IsValidEmail” adında bir method tanımlıyorum.

```
1 başvuru
88  private bool IsValidEmail(string email) // verilen parametrenin eposta olup olmadığını kontrol eder
89  {
90      if (string.IsNullOrWhiteSpace(email))
91          return true;
92      try
93      {
94          MailAddress mailAddress = new MailAddress(email);
95          return true;
96      }
97      catch (FormatException)
98      {
99          return false;
100     }
101 }
```

Textbox placeholder kodları :

```
114  1 başvuru
115  private void textBox1_Enter(object sender, EventArgs e)
116  {
117      if(txtEposta.Text=="E-Posta adresinizi giriniz")
118      {
119          txtEposta.Text = "";
120          txtEposta.ForeColor = Color.Black;
121      }
122
123  1 başvuru
124  private void textBox1_Leave(object sender, EventArgs e)
125  {
126      if(txtEposta.Text=="")
127      {
128          txtEposta.Text = "E-Posta adresinizi giriniz";
129          txtEposta.ForeColor = Color.Gray;
130      }

```

Textbox format kontrolü ve hata mesajı işlemleri,

```
133  1 başvuru
134  private void txtEposta_Validating(object sender, CancelEventArgs e)
135  {
136      string email = txtEposta.Text;
137      bool isValid = IsValidEmail(email);
138      if (!isValid)
139      {
140          if (txtEposta.Text == "E-Posta adresinizi giriniz")
141          {
142              txtEposta.BackColor = Color.White;
143              errorProvider.SetError(txtEposta, "");
144          }
145          else
146          {
147              // Eposta yanlış formatta
148              txtEposta.BackColor = Color.FromArgb(255, 114, 86);
149              txtEposta.ForeColor = Color.Black;
150              errorProvider.SetError(txtEposta, "Lütfen geçerli bir eposta adresi girin.");
151              e.Cancel = false;
152          }
153      }
154      else
155      {
156          // Eposta doğru formatlidir
157          txtEposta.BackColor = Color.White;
158          txtEposta.ForeColor = Color.Black;
159          errorProvider.SetError(txtEposta, "");
160          e.Cancel = false;
161      }
162  }
```

Sifre gönder butonuna tıklandığında çalışacak kodlar :

```
169  169  private void btnSifreGonder_Click(object sender, EventArgs e)
170  {
171      if (errorProvider.GetError(txtEposta) == "")
172      {
173          // Sifre Gönderme işlemi yapılmıyor.
174          if (txtEposta.Text == "E-Posta adresinizi giriniz")
175          {
176              MessageBox.Show("Lütfen bir E-Posta adresi girin.");
177              txtEposta.Focus();
178          }
179          else
180          {
181              SifreGonder(txtEposta.Text);
182          }
183      }
184      else
185      {
186          MessageBox.Show("Lütfen geçerli bir E-Posta adresi girin.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
187  }
```

E4- Araç Kiralama Ekranı

The screenshot shows a Windows application window titled "Araç Kiralama İşlemleri". On the left, there's a sidebar with links like "Merhaba , Emre", "Araç Kiralama İşlemleri", "Araç Teslim İşlemleri", "Araç Yönetimi", "Müşteri Yönetimi", "Hesap Ayarları", and "Çıkış Yap". Below the sidebar is a contact number "0216 555 44 32". The main area has a title "Kiralanabilir Araçlar" and several search filters: Plaka (26GD4521), Yılı (2013), Vites Türü (Manuel), Yakıt Türü (Dizel), Kiralama Şekli (Sezniz...), Alış Tarihi (23/05/2023), Müşteri (Sezniz...), and Teslim Tarihi (24/05/2023). A large table lists available cars with columns: Plaka, Marka, Model, Yılı, Vites Türü, Yakıt Türü, Günlük Ücret, Haftalık Ücret, and Aylık Ücret. The table contains 10 entries.

Plaka	Marka	Model	Yılı	Vites Türü	Yakıt Türü	Günlük Ücret	Haftalık Ücret	Aylık Ücret
26GD4521	Renault	Clio	2013	Manuel	Dizel	600,00	3780,00	15000,00
34CCZ3143	Renault	Clio	2013	Otomatik	Dizel	600,00	3780,00	15000,00
34CDA2252	Toyota	Corolla	2010	Manuel	Dizel	600,00	3780,00	15000,00
34TFP34	Renault	Clio	2013	Otomatik	Dizel	600,00	3780,00	15000,00
45DSA45	Renault	Megane	2015	Otomatik	Dizel	700,00	4550,00	18600,00
01AA1252	Toyota	Corolla	2012	Otomatik	Dizel	700,00	4550,00	18600,00
06DD5243	Volkswagen	Caddy	2011	Manuel	Dizel	700,00	4550,00	18600,00
77GF4698	Renault	Megane	2013	Manuel	Dizel	700,00	4550,00	18600,00
34DSA1242	Mercedes	C180	2010	Otomatik	Benzin	700,00	4550,00	18600,00

Kodları

Veritabanı bağlantısı, ErrorProvider nesnesinin oluşturulması ve form closing eventi kod blogu:

```
19  public MainScreen()
20  {
21      InitializeComponent();
22  }
23
24 // MSSQL Veritabanı bağlantısı kurar
25 SqlConnection bağlantı = new SqlConnection("Data Source=DESKTOP-BTIS91A;Initial Catalog=Db_AracKiralama;Integrated Security=True");
26 ErrorProvider errorProvider= new ErrorProvider();
27
28
29 private void MainScreen_FormClosing(object sender, FormClosingEventArgs e)
30 {
31     if (e.CloseReason == CloseReason.UserClosing) // Form, kullanıcı tarafından kapatıldıysa
32     {
33         DialogResult result = MessageBox.Show("Uygulamadan çıkmak istediğinizize emin misiniz?", "Uyarı", MessageBoxButtons.YesNo);
34
35         if (result == DialogResult.Yes)
36         {
37             e.Cancel = false;
38
39             if (this.DialogResult != DialogResult.OK) // Eğer form dialog olarak kullanılmıyorsa
40             {
41                 // Login ekranına yönlendir
42                 LoginScreen loginScreen = new LoginScreen();
43                 loginScreen.Show();
44             }
45         }
46         else
47         {
48             e.Cancel = true; // Formun kapatılmasını engelle
49         }
50     }
51     else // Diğer kapatılma sebeplerinde uygulamayı kapat
52     {
53         Application.Exit();
54     }
55 }
```

Çıkış yap labelına tıklandığında çalışacak kod bloğu ve load eventi,

```
59 1 reference
60 private void LnkLabelCikis_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
61 {
62     // Çıkış yap labelına tıklandığında dialog oluşturulması işlemi
63     DialogResult result = MessageBox.Show("Hesabınızdan çıkmak istediğinizden emin misiniz?", "Uyarı", MessageBoxButtons.YesNo);
64
65     if (result == DialogResult.Yes)
66     {
67
68         if (this.DialogResult != DialogResult.OK) // Eğer form dialog olarak kullanılmıyorsa
69         {
70             // Login ekranına yönlendir
71             LoginScreen loginScreen = new LoginScreen();
72             loginScreen.Show();
73             this.Hide();
74         }
75     }
76
77 }
78
79 1 reference
80 private void MainScreen_Load(object sender, EventArgs e)
81 {
82
83     // Karşılama Mesajı
84     lblLeftName.Text = "Merhaba , " + LoginScreen.userName;
85
86     dateTimePickerLoad(); // timepicker getDate() işlemi ve format işlemi
87     comboBoxLoad(); // combobox için belirlenen indexlerin set edilmesi
88
89
90     // verileri çek
91     getCarInfo();
92     getCustomersNameSurname();
93
94
95 }
```

getCustomersNameSurname() methodu içerisindeki kodlar,

```
96 1 reference
97 private void getCustomersNameSurname() // veritabanından müşterilerin ad - soyadını çekme işlemi
98 {
99
100    try
101    {
102        baglanti.Open(); // bağlantıının açılması
103        SqlCommand cmd = new SqlCommand("getCustomerNameSurname", baglanti); // sqlCommand oluşturulması
104        cmd.CommandType = CommandType.StoredProcedure; // command türünün storedProcedure olarak belirlenmesi
105
106        SqlDataReader reader = cmd.ExecuteReader(); // sqlCommandın çalıştırılıp dönen değerin reader ile okunması
107        if(reader.HasRows)
108        {
109            while(reader.Read()) // reader içerisindeki verilerin müşteri combobox'ına eklenmesi
110            {
111                cmbBoxMusteri.Items.Add(reader.GetString(1));
112            }
113        }
114    }
115    catch (Exception ex)
116    {
117        MessageBox.Show(ex.Message);
118    }
119    finally
120    {
121        baglanti.Close(); // bağlantıının kapatılması
122    }
123 }
```

getSelectedCarInfo() ve getCarInfo() methodları içerisindeki kodlar,

```
124 1 reference
125 private void getSelectedCarInfo() // seçili datagridview deki verileri textboxlara yazdırma
126 {
127     if (dataAraclar.SelectedRows.Count > 0)
128     {
129         DataGridViewRow selectedRow = dataAraclar.SelectedRows[0];
130         if (selectedRow.Cells[0].Value != null) // seçili satır varsa
131         {
132             txtPlakaInfo.Text = selectedRow.Cells[0].Value.ToString();
133             txtMarkaInfo.Text = selectedRow.Cells[1].Value.ToString();
134             txtModelInfo.Text = selectedRow.Cells[2].Value.ToString();
135             txtVitesInfo.Text = selectedRow.Cells[3].Value.ToString();
136             txtYakitInfo.Text = selectedRow.Cells[4].Value.ToString();
137             txtYakitInfo.Text = selectedRow.Cells[5].Value.ToString();
138         }
139     }
140 2 references
141 private void getCarInfo() // araçları veritabanından çekme işlemi
142 {
143     if(baglanti.State == ConnectionState.Open)
144     { baglanti.Close(); }

145     DataTable dataTable= new DataTable(); // DataTable nesnesini oluşturma
146     try
147     {
148         baglanti.Open(); // bağlantıyı aç
149         SqlCommand cmd = new SqlCommand("getCarData", baglanti); // sqlCommand oluşturma
150         cmd.CommandType = CommandType.StoredProcedure; // command türünü storedProcedure olarak belirleme
151
152         SqlDataReader reader = cmd.ExecuteReader(); // execute işlemi ve dönen verinin reader nesnesine alınması
153         dataTable.Load(reader); // reader içerisindeki verinin dataTable'a aktarılması
154         dataAraclar.DataSource= dataTable; // datagridview e dataTable daki verilerin aktarımı
155
156     }
157     catch(Exception ex)
158     {
159         MessageBox.Show(ex.Message);
160     }
161     finally { baglanti.Close(); } // bağlantıyı kapat
162 }
```

dateTimePickerLoad() ve comboBoxLoad() methodları içerisindeki kodlar,

```
163 1 reference
164 private void dateTimePickerLoad() //timepicker format ayarları ve minDate belirleme
165 {
166     timePickerAlis.CustomFormat = "dd/MM/yyyy";
167     timePickerTeslim.CustomFormat = "dd/MM/yyyy";
168     timePickerAlis.MinDate = DateTime.Today;
169     timePickerTeslim.MinDate = timePickerAlis.Value.AddDays(1); // teslim tarihinin en az 1 gün sonra olmasının ayarlanması
170
171 1 reference
172 private void comboBoxLoad() // load olayında comboboxların seçili indexlerinin ayarlanması
173 {
174     //Combobox SelectedIndex
175     cmbBoxKiralamaSekli.SelectedIndex = 0;
176     cmbBoxMusteri.SelectedIndex = 0;
177 }
```

timePickerAlis ValueChanged eventi kodbloğu,

```
255 1 reference
256 private void timePickerAlis_ValueChanged(object sender, EventArgs e)
257 {
258     timePickerTeslim.MinDate = timePickerAlis.Value.AddDays(1);
259     if (timePickerAlis.Value < DateTime.Today)
260     {
261         errorProvider.SetError(timePickerAlis, "Geçmiş bir tarih seçemezsiniz.");
262         timePickerAlis.Value = DateTime.Today;
263     }
264     else
265     {
266         errorProvider.SetError(timePickerAlis, "");
267     }
268 }
```

“Aracı Kirala” butonuna tıklandığında gerçekleşecek işlemler,

```
1 reference
178 private void btnAraciKirala_Click(object sender, EventArgs e) //Aracı Kirala butonuna tıklandığında yapılacak işlemler
{
    DialogResult result =
        MessageBox.Show(txtPlakaInfo.Text + " Plakalı "
            + txtMarkaInfo.Text + " marka araç "
            + cmbBoxMusteri.SelectedItem.ToString()
            + " adına kiralananacaktır. Devam etmek istiyor musunuz ?", "Uyarı", MessageBoxButtons.YesNo);

    if (result == DialogResult.Yes)
    {
        DateTime dateAlis = DateTime.ParseExact(timePickerAlis.Text, "dd/MM/yyyy", CultureInfo.InvariantCulture);
        DateTime dateTeslim = DateTime.ParseExact(timePickerTeslim.Text, "dd/MM/yyyy", CultureInfo.InvariantCulture);
        string dateAlisSqlFormat = dateAlis.ToString("yyyy-MM-dd"); // SQL format dönüştürme
        string dateTeslimSqlFormat = dateTeslim.ToString("yyyy-MM-dd"); // SQL formatına dönüştürme

        // hata kontrolü
        if (errorProvider.GetError(cmbBoxKiralamaSekli) == "" & errorProvider.GetError(cmbBoxMusteri) == "")
        {
            //seçilmemiş alan var mı
            if (cmbBoxKiralamaSekli.SelectedItem.ToString() == "Seçiniz..." || cmbBoxMusteri.SelectedItem.ToString() == "Seçiniz...")
            {
                MessageBox.Show("Lütfen eksik bilgileri doldurun");
            }
            else
            {
                // Veritabanı işlemleri
                try
                {
                    if (baglanti.State == ConnectionState.Open)
                    {
                        baglanti.Close();
                    }
                    baglanti.Open(); // bağlantıyi aç
                    SqlCommand cmd = new SqlCommand("setRentedTest", baglanti); // testt!!!
                    cmd.CommandType = CommandType.StoredProcedure;
                    // storedProcedure parametre geçme
                    cmd.Parameters.AddWithValue("@paramPlaka", txtPlakaInfo.Text);
                    cmd.Parameters.AddWithValue("@paramMusteri", cmbBoxMusteri.SelectedItem.ToString());
                    cmd.Parameters.AddWithValue("@paramKiralamaTuru", cmbBoxKiralamaSekli.SelectedItem.ToString());
                    cmd.Parameters.AddWithValue("@paramAlisTarihi", dateAlisSqlFormat);
                    cmd.Parameters.AddWithValue("@paramTeslimTarihi", dateTeslimSqlFormat);
                    cmd.ExecuteNonQuery(); // command execute işlemi
                    MessageBox.Show(txtPlakaInfo.Text + " Plakalı "
                        + txtMarkaInfo.Text
                        + " marka araç "
                        + cmbBoxMusteri.SelectedItem.ToString()
                        + " adına kiralandı.");
                    getCarInfo(); // datagridview yenile
                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.Message);
                }
                finally
                {
                    baglanti.Close(); // bağlantıyi kapat
                }
            }
        }
        else // eksik bilgi var ise
        {
            MessageBox.Show("Lütfen eksik bilgileri doldurun");
        }
    }
}
```

Datagridview seçili satırın değişmesi sonucu çalışacak kod bloğu,

```
1 reference
246 private void dataAraclar_SelectionChanged(object sender, EventArgs e)
{
    // Seçili aracın bilgilerini ilgili textlere yazdırma
    getSelectedCarInfo();
}
```

Alış tarihi seçildiğinde çalışacak kod blogu,

```
253 1 reference
254 private void timePickerAlis_ValueChanged(object sender, EventArgs e) // Alış tarihi seçildiğinde yapılacak kontroller
255 {
256     timePickerTeslim.MinDate = timePickerAlis.Value.AddDays(1); // Teslim tarihinin min değeri alış tarihinden 1 fazla olarak ayarlandı
257     if (timePickerAlis.Value < DateTime.Today)
258     {
259         errorProvider.SetError(timePickerAlis, "Geçmiş bir tarih seçemezsiniz.");
260         timePickerAlis.Value = DateTime.Today;
261     }
262     else
263     {
264         errorProvider.SetError(timePickerAlis, "");
265     }
266 }
```

Datagridview RowPrePaint eventi kodları,

```
284 1 reference
285 private void dataAraclar_RowPrePaint(object sender, DataGridViewRowPrePaintEventArgs e)
286 {
287     // Eğer satırın indexi tek sayı ise arkaplan rengini gri olarak ayarla
288     if (e.RowIndex % 2 == 0)
289     {
290         dataAraclar.Rows[e.RowIndex].DefaultCellStyle.BackColor = Color.White;
291     }
292     // Değilse arkaplan rengini beyaz olarak ayarla
293     else
294     {
295         dataAraclar.Rows[e.RowIndex].DefaultCellStyle.BackColor = ColorTranslator.FromHtml("#e5e5e5");
296     }
297 }
```

Menü labellarının click eventleri,

```
327 1 reference
328 private void LnkLabelAracTeslim_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e) // araç teslim ekranını açar
329 {
330     this.Hide();
331     VehicleDeliveryScreen vehicleDeliveryScreen = new VehicleDeliveryScreen();
332     vehicleDeliveryScreen.Show();
333 }
334 1 reference
335 private void LnkLabelAracYonetim_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e) // araç yönetim ekranını açar
336 {
337     this.Hide();
338     VehicleManagementScreen vehicleManagementScreen = new VehicleManagementScreen();
339     vehicleManagementScreen.Show();
340 }
341 1 reference
342 private void LnkLabelMusteriYonetim_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e) // müşteri yönetim ekranını açar
343 {
344     this.Hide();
345     CustomerManagementScreen customerManagementScreen = new CustomerManagementScreen();
346     customerManagementScreen.Show();
347 }
348 1 reference
349 private void LnkLabelHesapAyar_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e) // hesap ayarları ekranını açar
350 {
351     this.Hide();
352     AccountSettingsScreen accountSettingsScreen = new AccountSettingsScreen();
353     accountSettingsScreen.Show();
354 }
```

E5-Araç Teslim Ekranı

Plaka	Marka	Model	Yılı	Vites Türü	Yakıt Türü	Kiralama Şekli	Müşteri Adı	Toplam Borç	Alış Tarihi	Teslim Tarihi	Durumu
45DSA45	Renault	Megane	2015	Otomatik	Dizel	Aylık	Emre	19220.00	25.05.2023	25.06.2023	True

Kodları

Veritabanı bağlantısı , ErrorProvider nesnesinin bağlanması , form çağrıldığında çalışacak methodlar ve getDeliveryData() methodu:

```
16 // veritabanı bağlantısı
17 SqlConnection bağlantı = new SqlConnection("Data Source=DESKTOP-BTIS91A;Initial Catalog=Db_Arakiralama;Integrated Security=True");
18 // hata yönetimi
19 ErrorProvider errorProvider = new ErrorProvider();
20
21 public VehicleDeliveryScreen()
22 {
23     InitializeComponent();
24
25     // Karşılama Mesajı
26     lblLeftName.Text = "Merhaba , " + LoginScreen.userName;
27
28     // Veri çekme
29     getDeliveryData();
30     getSelectedCarInfo();
31 }
32
33 private void getDeliveryData() // kiradaki araçların datasını çeker
34 {
35     if (baglanti.State == ConnectionState.Open)
36     { bağlantı.Close(); }
37
38     DataTable dataTable = new DataTable(); //DataTable nesnesinin oluşturulması
39     try
40     {
41         bağlantı.Open(); // bağlantıyi aç
42         SqlCommand cmd = new SqlCommand("getDeliveryDetails", bağlantı); // sqlCommand oluşturma
43         cmd.CommandType = CommandType.StoredProcedure; // command türünü storedProcedure olarak belirleme
44
45         SqlDataReader reader = cmd.ExecuteReader(); // execute işlemi ve dönen verinin reader nesnesinde tutulması
46         dataTable.Load(reader); // reader nesnesindeki verinin dataTable'a yüklenmesi
47         dataAracTeslim.DataSource = dataTable; // datagridviewe dataTable nesnesindeki verileri gösterme
48
49     }
50     catch (Exception ex)
51     {
52         MessageBox.Show(ex.Message);
53     }
54     finally { bağlantı.Close(); } // bağlantıyi kapat
55 }
```

getSelectedCarInfo() methodu içerisindeki kodlar,

```
59  2 references
60  private void getSelectedCarInfo() // datagridview de seçili olan aracın bilgilerini textboxlara yazdırma
61  {
62      if (dataAracTeslim.SelectedRows.Count > 0)
63      {
64          DataGridViewRow selectedRow = dataAracTeslim.SelectedRows[0];
65          if (selectedRow.Cells[0].Value != null)
66          {
67              txtPlakaInfo.Text = selectedRow.Cells[0].Value.ToString();
68              txtMarkaInfo.Text = selectedRow.Cells[1].Value.ToString();
69              txtModelInfo.Text = selectedRow.Cells[2].Value.ToString();
70              txtYilInfo.Text= selectedRow.Cells[3].Value.ToString();
71              txtVitesTuruInfo.Text= selectedRow.Cells[4].Value.ToString();
72              txtYakitTuruInfo.Text= selectedRow.Cells[5].Value.ToString();
73              txtKiralamaSekliInfo.Text = selectedRow.Cells[6].Value.ToString();
74              txtMusteriAdiInfo.Text = selectedRow.Cells[7].Value.ToString();
75              txtBorcInfo.Text = selectedRow.Cells[8].Value.ToString();
76              txtDurum.Text = selectedRow.Cells[11].Value.ToString();
77          }
78      }
79  }
```

FormClosing eventi kod blogu,

```
81  1 reference
82  private void VehicleDeliveryScreen_FormClosing(object sender, FormClosingEventArgs e)
83  {
84      if (e.CloseReason == CloseReason.UserClosing) // Form, kullanıcı tarafından kapatıldıysa
85      {
86          DialogResult result = MessageBox.Show("Uygulamadan çıkmak istediğiniz emin misiniz?", "Uyarı", MessageBoxButtons.YesNo);
87
88          if (result == DialogResult.Yes)
89          {
90              e.Cancel = false;
91
92              if (this.DialogResult != DialogResult.OK) // Eğer form dialog olarak kullanılmıyorsa
93              {
94                  // Login ekranına yönlendir
95                  LoginScreen loginScreen = new LoginScreen();
96                  loginScreen.Show();
97              }
98          }
99          else
100         {
101             e.Cancel = true; // Formun kapatılmasını engelle
102         }
103     }
104     else // Diğer kapatılma sebeplerinde uygulamayı kapat
105     {
106         Application.Exit();
107     }
108 }
```

Textbox Placeholder kodları,

```
-- 119 // Placeholder kodları
120 1 reference
121 private void txtGecikenGun_Enter(object sender, EventArgs e)
122 {
123     if (txtGecikenGun.Text == "Geciken Gün")
124     {
125         txtGecikenGun.Text = "";
126         txtGecikenGun.ForeColor = Color.Black;
127     }
128
129 1 reference
130 private void txtGecikenGun_Leave(object sender, EventArgs e)
131 {
132     if (txtGecikenGun.Text == "")
133     {
134         txtGecikenGun.Text = "Geciken Gün";
135         txtGecikenGun.ForeColor = Color.Gray;
136     }
137 }
```

```

138     1 reference
139     private void txtHasarUcreti_Enter(object sender, EventArgs e)
140     {
141         if (txtHasarUcreti.Text == "Hasar Ücreti")
142         {
143             txtHasarUcreti.Text = "";
144             txtHasarUcreti.ForeColor = Color.Black;
145         }
146     }
147     1 reference
148     private void txtHasarUcreti_Leave(object sender, EventArgs e)
149     {
150         if (txtHasarUcreti.Text == "")
151         {
152             txtHasarUcreti.Text = "Hasar Ücreti";
153             txtHasarUcreti.ForeColor = Color.Gray;
154         }
155     }
156     1 reference
157     private void txtCезаUcreti_Leave(object sender, EventArgs e)
158     {
159         if (txtCезаUcreti.Text == "")
160         {
161             txtCезаUcreti.Text = "Ceza Ücreti";
162             txtCезаUcreti.ForeColor = Color.Gray;
163         }
164     }
165     1 reference
166     private void txtCезаUcreti_Enter(object sender, EventArgs e)
167     {
168         if (txtCезаUcreti.Text == "Ceza Ücreti")
169         {
170             txtCезаUcreti.Text = "";
171             txtCезаUcreti.ForeColor = Color.Black;
172         }
173     }

```

“Çıkış Yap” butonuna tıklandığında çalışacak kod bloğu,

```

246     // çıkış yap labelına tıklandığında çalışacak event
247     1 reference
248     private void LnkLabelCikis_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
249     {
250         DialogResult result = MessageBox.Show("Hesabınızdan çıkmak istediğiniz emin misiniz?", "Uyarı", MessageBoxButtons.YesNo);
251         if (result == DialogResult.Yes)
252         {
253
254             if (this.DialogResult != DialogResult.OK) // Eğer form dialog olarak kullanılmiyorsa
255             {
256                 // Login ekranına yönlendir
257                 LoginScreen loginScreen = new LoginScreen();
258                 loginScreen.Show();
259                 this.Hide();
260             }
261         }
262     }
263 }

```

Datagridview SelectionChanged eventi kodbloğu,

```

174     1 reference
175     private void dataAracTeslim_SelectionChanged(object sender, EventArgs e)
176     {
177         // Seçilen arabanın bilgilerini getir.
178         getSelectedCarInfo();
179     }

```

“Aracı Teslim Et” butonuna tıklandığında çalışacak kod blogu,

```
1 reference
180     private void btnAracTeslim_Click(object sender, EventArgs e)
181     {
182         if (txtPlakaInfo.Text != "")
183         {
184             int gecikenGun = 0;
185             int cezaUcreti = 0;
186             int hasarUcreti = 0;
187             Boolean liste = false;
188             if (txtGecikenGun.Text != "Geciken Gün")
189             {
190                 gecikenGun += Convert.ToInt32(txtGecikenGun.Text);
191             }
192             if (txtCezaUcreti.Text != "Ceza Ücreti")
193             {
194                 cezaUcreti += Convert.ToInt32(txtCezaUcreti.Text);
195             }
196             if (txtHasarUcreti.Text != "Hasar Ücreti")
197             {
198                 hasarUcreti += Convert.ToInt32(txtHasarUcreti.Text);
199             }
200             if (chkListele.Checked)
201             {
202                 liste = true;
203             }
204         try
205         {
206
207             if (baglanti.State == ConnectionState.Closed)
208             { baglanti.Open(); } // bağlantıyı aç
209
210             SqlCommand cmd = new SqlCommand("updateDeliveryDetails", baglanti); // command oluşturma
211             cmd.CommandType = CommandType.StoredProcedure; // command türü storedProcedure olarak ayarlama
212             // parametre geçiriliyor
213             cmd.Parameters.AddWithValue("@paramPlaka", txtPlakaInfo.Text);
214             cmd.Parameters.AddWithValue("@paramGecikenGun", gecikenGun);
215             cmd.Parameters.AddWithValue("@paramCezaUcreti", cezaUcreti);
216             cmd.Parameters.AddWithValue("@paramHasarUcreti", hasarUcreti);
217             cmd.Parameters.AddWithValue("@paramListele", liste);
218             cmd.ExecuteNonQuery(); // execute işlemi
219             MessageBox.Show(txtPlakaInfo.Text + " Plakalı araç teslim alındı.");
220             getDeliveryData(); // datagridview refresh işlemi
221
222             // İşlem sonrası placeholder
223             txtGecikenGun.ForeColor = Color.Gray;
224             txtCezaUcreti.ForeColor = Color.Gray;
225             txtHasarUcreti.ForeColor = Color.Gray;
226             txtGecikenGun.Text = "Geciken Gün";
227             txtHasarUcreti.Text = "Hasar Ücreti";
228             txtCezaUcreti.Text = "Ceza Ücreti";
229
230         }
231         catch (Exception ex)
232         {
233             MessageBox.Show(ex.Message);
234         }
235         finally
236         {
237             baglanti.Close(); // bağlantıyı kapat
238         }
239     }
240     else
241     {
242         MessageBox.Show("Lütfen teslim almak istediğiniz aracı seçiniz.");
243     }
244 }
```

Ekran değişimi için kullandığım LinkLabel'ların click eventleri,

```
265 // menü ekran geçişleri
266 1 reference
267 private void LnkLabelArackiralamaIslemleri_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
268 {
269     this.Hide();
270     MainScreen mainScreen = new MainScreen();
271     mainScreen.Show();
272 }
273 1 reference
274 private void LnkLabelAracYonetim_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
275 {
276     this.Hide();
277     VehicleManagmentScreen vehicleManagementScreen = new VehicleManagementScreen();
278     vehicleManagementScreen.Show();
279 }
280 1 reference
281 private void LnkLabelMusteriYonetim_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
282 {
283     this.Hide();
284     CustomerManagementScreen customerManagementScreen = new CustomerManagementScreen();
285     customerManagementScreen.Show();
286 }
287 1 reference
288 private void LnkLabelHesapAyar_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
289 {
290     AccountSettingsScreen accountSettingsScreen = new AccountSettingsScreen();
291     accountSettingsScreen.Show();
292     this.Hide();
293 }
```

E6-Araç Yönetimi Ekranı

Plaka	Marka	Model	Yılı	Vites Türü	Yakıt Türü	Segment	Günlük Ücret	Haftalık Ücret	Aylık Ücret	Durumu
34ABC34	Audi	A6	2015	Manuel	Dizel	E	900,00	5950,00	25000,00	True
45DSA45	Renault	Megane	2015	Otomatik	Dizel	C	700,00	4550,00	18600,00	False
48DCA42	Bmw	520d	2013	Manuel	Dizel	E	900,00	5950,00	25000,00	True
01AA1252	Toyota	Corolla	2012	Otomatik	Dizel	C	700,00	4550,00	18600,00	True
06DD5243	Volkswagen	Caddy	2011	Manuel	Dizel	C	700,00	4550,00	18600,00	True
52EF4521	Volkswagen	Passat	2016	Manuel	Benzin	D	800,00	5250,00	21000,00	True
26GD4521	Renault	Clio	2013	Manuel	Dizel	B	600,00	3780,00	15000,00	True
12TT4231	Mercedes	C180	2011	Otomatik	Benzin	D	800,00	5250,00	21000,00	True
44TS2154	Mercedes	Vito	2012	Otomatik	Dizel	D	800,00	5250,00	21000,00	True

Kodları

Veritabanı bağlantısı , ErrorProvider nesnesinin bağlanması ve form çağrıldığında çalışacak methodlar:

```
19 //Veritabanı bağlantısı
20
21 SqlConnection bağlantı = new SqlConnection("Data Source=DESKTOP-BTIS91A;Initial Catalog=Db_Arakkiralama;Integrated Security=True");
22
23 //Hata yönetim aracı
24 ErrorProvider errorProvider = new ErrorProvider();
25
26 string oldPlaka = "";
27 4 references
28 public VehicleManagementScreen()
29 {
30
31     InitializeComponent();
32
33     // Karşılama Mesajı
34     lblLeftName.Text = "Merhaba , " + LoginScreen.userName;
35
36     //Combobox Load
37     comboBoxLoad();
38
39     //Verileri Çek
40     getAllCarsData();
41     getSelectedCarInfo();
42     getBrand();
43     getSegment();
}
```

getAllCarsData() methodu içerisindeki kodlar,

```
46  | 4 references
47  |  private void getAllCarsData() // veritabanından araçları çeker
48  |  {
49  |  if (baglanti.State == ConnectionState.Open)
50  |  { baglanti.Close(); }
51  |
52  |  DataTable dataTable = new DataTable(); // DataTable nesnesi oluşturma
53  |  try
54  |  {
55  |  baglanti.Open(); // bağlantıyi aç
56  |  SqlCommand cmd = new SqlCommand("getCarsAllData", baglanti); // sqlCommand oluşturma
57  |  cmd.CommandType = CommandType.StoredProcedure; // command türünü storedProcedure olarak ayarlama
58  |  SqlDataReader reader = cmd.ExecuteReader(); // execute işlemi ve dönen verinin reader nesnesine yazılması
59  |  dataTable.Load(reader); // reader içerisindeki verinin dataTable a yüklenmesi
60  |  reader.Close(); // reader nesnesinin kapatılması
61  |  dataAraclar.DataSource = dataTable; // dataTable içerisindeki verinin datagridviewe aktarılması
62  |
63  |  }
64  |  catch (Exception ex)
65  |  {
66  |  MessageBox.Show(ex.Message);
67  |  }
68  |  finally { baglanti.Close(); } // bağlantıyi kapat
```

getBrand() methodu içerisindeki kodlar,

```
70  | 1 reference
71  |  private void getBrand() // veritabanından markaları çekmek için
72  |  {
73  |  if (baglanti.State == ConnectionState.Open)
74  |  { baglanti.Close(); }
75  |
76  |  try
77  |  {
78  |  baglanti.Open(); // bağlantıyi aç
79  |  string query = "select Marka from tblMarka"; // sql sorgu cümlesi
80  |  SqlCommand cmd = new SqlCommand(query, baglanti); // sqlCommand oluşturma
81  |  SqlDataReader reader = cmd.ExecuteReader(); // execute işlemi ve dönen verinin reader'a aktarılması
82  |  List<string> markalar = new List<string>();
83  |  while (reader.Read())
84  |  {
85  |  markalar.Add(reader.GetString(0)); // markalar listesine reader içindeki verinin eklenmesi
86  |  }
87  |  markalar.Sort(); // markaları sıralama
88  |
89  |  foreach (var item in markalar)
90  |  {
91  |  cmbBoxMarka.Items.Add(item); // sıralanmış listeyi comboBox içeresine ekleme
92  |  }
93  |  reader.Close(); // reader nesnesinin kapatılması
94  |
95  |  }
96  |  catch (Exception ex)
97  |  {
98  |  MessageBox.Show(ex.Message);
99  |  }
100 |  finally { baglanti.Close(); } // bağlantıyi kapat
```

getSelectedCarInfo() methodu içerisindeki kodlar,

```
102 | 3 references
103 |  private void getSelectedCarInfo() // seçili araç bilgisini getirir
104 |  {
105 |  if (dataAraclar.SelectedRows.Count > 0)
106 |  {
107 |  DataGridViewRow selectedRow = dataAraclar.SelectedRows[0];
108 |  if (selectedRow.Cells[0].Value != null)
109 |  {
110 |  txtPlakaInfo.Text = selectedRow.Cells[0].Value.ToString();
111 |  txtMarkaInfo.Text = selectedRow.Cells[1].Value.ToString();
112 |  txtModelInfo.Text = selectedRow.Cells[2].Value.ToString();
113 |  txtYilInfo.Text = selectedRow.Cells[3].Value.ToString();
114 |  txtVitesTurInfo.Text = selectedRow.Cells[4].Value.ToString();
115 |  txtYakitTurInfo.Text = selectedRow.Cells[5].Value.ToString();
116 |  txtGunlukUcret.Text = selectedRow.Cells[7].Value.ToString();
117 |  txtHaftalıkUcret.Text = selectedRow.Cells[8].Value.ToString();
118 |  txtAylıkUcret.Text = selectedRow.Cells[9].Value.ToString();
119 |  txtDurumuInfo.Text = selectedRow.Cells[10].Value.ToString();
120 |  }
121 |  }
```

setEnableTextbox() ve setDisableTextbox() methodları içerisindeki kodlar,

```
122  1 reference
123  private void setEnableTextbox() // ilgili textboxları enable yapar
124  {
125      txtPlakaInfo.Enabled = true;
126      txtPlakaInfo.TextAlign = HorizontalAlignment.Left;
127      txtPlakaInfo.ForeColor = Color.Black;
128      //txtModelInfo.Enabled = true;
129      //txtMarkaInfo.Enabled = true;
130      //txtVitesTurInfo.Enabled = true;
131      txtYilInfo.Enabled = true;
132      txtYilInfo.TextAlign = HorizontalAlignment.Left;
133      txtYilInfo.ForeColor = Color.Black;
134      //txtYakitTurInfo.Enabled = true;
135  }
136  2 references
137  private void setDisableTextbox() // ilgili textboxları disable yapar
138  {
139      txtPlakaInfo.Enabled = false;
140      txtPlakaInfo.TextAlign = HorizontalAlignment.Center;
141      txtPlakaInfo.ForeColor = Color.Gray;
142      //txtModelInfo.Enabled = false;
143      //txtMarkaInfo.Enabled = false;
144      //txtVitesTurInfo.Enabled = false;
145      txtYilInfo.Enabled = false;
146      txtYilInfo.TextAlign = HorizontalAlignment.Center;
147      txtYilInfo.ForeColor = Color.Gray;
148      //txtYakitTurInfo.Enabled = false;
149  }
```

getSegment() methodu içerisindeki kodlar,

```
150  1 reference
151  private void getSegment() // veritabanından segmentleri çeker
152  {
153      if (baglanti.State == ConnectionState.Open)
154      { baglanti.Close(); }
155
156      try
157      {
158          baglanti.Open(); // bağlantıyi aç
159          string query = "select Segment from tblSegment"; // sql sorgu cümlesi
160          SqlCommand cmd = new SqlCommand(query, baglanti); // sqlCommand oluşturma
161          SqlDataReader reader = cmd.ExecuteReader(); // execute işlemi ve reader nesnesine dönen verinin atılması
162          List<string> segmentler = new List<string>();
163          while (reader.Read())
164          {
165              segmentler.Add(reader.GetString(0)); // reader içerisindeki veriyi segmentler listesine ekleme
166          }
167          segmentler.Sort(); // listeyi sıralama
168
169          foreach (var item in segmentler)
170          {
171              cmbBoxSegment.Items.Add(item); // sıralanmış listeyi segment comboboxına ekleme
172          }
173          reader.Close(); // reader kapat
174
175      }
176      catch (Exception ex)
177      {
178          MessageBox.Show(ex.Message);
179      }
180  }
```

comboBoxLoad() methodu içerisindeki kodlar,

```
181  1 reference
182  private void comboBoxLoad() // formun load olayında kullanmak için , ilgili comboboxların varsayılan değerlerini belirler
183  {
184      //Combobox SelectedIndex
185      cmbBoxMarka.SelectedIndex = 0;
186      cmbBoxModel.SelectedIndex = 0;
187      cmbBoxDurum.SelectedIndex = 0;
188      cmbBoxVitesTuru.SelectedIndex = 0;
189      cmbBoxYakitTuru.SelectedIndex = 0;
190  }
```

getModel() methodu içerisindeki kodlar,

```
1 reference
192     private void getModel() // veritabanından modelleri çeker
193     {
194         if (baglanti.State == ConnectionState.Closed)
195             { baglanti.Open(); }
196
197         try
198         {
199             SqlCommand cmd = new SqlCommand("getMarkaId",baglanti); // markaId almak için sqlCommand oluşturma
200             cmd.CommandType= CommandType.StoredProcedure; // komut türünü stored procedure olarak ayarlama
201             // parametre tanımlanıyor..
202             SqlParameter paramMarkaAdi = new SqlParameter("@paramMarkaAdi", cmbBoxMarka.SelectedItem.ToString());
203             //parametre input olarak ayarlanıyor..
204             paramMarkaAdi.Direction = ParameterDirection.Input;
205             // parametre geçiriliyor..
206             cmd.Parameters.Add(paramMarkaAdi);
207             //dönen output değeri alınıyor..
208             SqlParameter paramId = new SqlParameter("@Id", SqlDbType.Int);
209             paramId.Direction = ParameterDirection.Output;
210             cmd.Parameters.Add(paramId); // id değerini commanda parametre geçme
211
212             // SqlCommand nesnesini çalıştırma
213             cmd.ExecuteNonQuery();
214
215             // Çıktı parametresindeki değeri alma
216             int markaId = (int)cmd.Parameters["@Id"].Value;
217             string query = "select ModelAdi from tblModel where MarkaId = @markaId "; // sql sorgusu oluşturma
218             SqlCommand cmdModel = new SqlCommand(query, baglanti); // sqlCommand oluşturma
219             cmdModel.Parameters.AddWithValue("@markaId",markaId); // parametre geçiriliyor
220             SqlDataReader reader = cmdModel.ExecuteReader(); // execute işlemi ve dönen verinin readera aktarımı
221             List<string> modeller = new List<string>(); // reader içerisindeki modelleri eklemek için liste(sıralama için)
222             if(reader.HasRows)
223             {
224                 while(reader.Read())
225                 {
226                     modeller.Add(reader.GetString(0)); // reader içerisindeki verinin listeye eklenmesi
227                 }
228                 modeller.Sort(); // modeller listesinin sıralanması
229             }
230             cmbBoxModel.Items.Clear(); // comboboxın içerisindeki temizlenmesi
231             cmbBoxModel.Items.Add("Seçiniz..."); //cmbBoxModel.SelectedIndex= 0;
232             foreach(var model in modeller)
233             {
234                 cmbBoxModel.Items.Add(model); // sıralanmış listeyi combobox a ekleme
235             }
236             reader.Close(); // reader kapatma
237
238         }
239         catch (Exception ex)
240         {
241
242             MessageBox.Show(ex.Message);
243         }
244         finally { baglanti.Close(); } // bağlantıyi kapat
245     }
246 }
```

Datagridview SelectionChanged eventi kodbloğu,

```
1 reference
448     private void dataAraclar_SelectionChanged(object sender, EventArgs e)
449     {
450         // Seçili araç bilgilerini çek
451         getSelectedCarInfo();
452     }
```

getModelForEdit() methodu içerisindeki kodlar,

```
1 reference
248     private void getModelForEdit() // modelleri getirir
249     {
250
251         if (baglanti.State == ConnectionState.Closed)
252             { baglanti.Open(); }
253
254         try
255         {
256             SqlCommand cmd = new SqlCommand("getMarkaId", baglanti); // sqlCommand oluşturma
257             cmd.CommandType = CommandType.StoredProcedure; // command türünü storedProcedure olarak belirleme
258             // parametre oluşturma
259             SqlParameter paramMarkaAdi = new SqlParameter("@paramMarkaAdi", cmbBoxMarkaEdit.SelectedItem.ToString());
260             paramMarkaAdi.Direction = ParameterDirection.Input; // parametre türünü input olarak ayarlar
261             cmd.Parameters.Add(paramMarkaAdi); // parametreyi ekler
262
263             SqlParameter paramId = new SqlParameter("@Id", SqlDbType.Int); // parametre oluşturma
264             paramId.Direction = ParameterDirection.Output; // parametre türünü output olarak ayarlama
265             cmd.Parameters.Add(paramId); // parametre ekleme
266
267             // SqlCommand nesnesini çalıştırma
268             cmd.ExecuteNonQuery();
269
270             // Çıktı parametresindeki değeri alma
271             int markaId = (int)cmd.Parameters["@Id"].Value;
272             string query = "select ModelAdi from tblModel where MarkaId = @markaId "; // sql sorgusu oluşturma
273             SqlCommand cmdModel = new SqlCommand(query, baglanti); // sqlCommand oluşturma
274             cmdModel.Parameters.AddWithValue("@markaId", markaId); // parametre geçiriliyor
275             SqlDataReader reader = cmdModel.ExecuteReader(); // execute işlemi ve reader nesnesine dönen verinin aktarımı
276             List<string> modeller = new List<string>(); // modeller listesinin oluşturulması (sıralama için)
277             if (reader.HasRows)
278             {
279                 while (reader.Read())
280                 {
281                     modeller.Add(reader.GetString(0)); // reader içerisindeki verinin listeye eklenmesi
282                 }
283                 modeller.Sort(); // modeller listesinin sıralanması
284             }
285             cmbBoxModelEdit.Items.Clear();
286             if(cmbBoxMarkaEdit.SelectedItem.ToString() == txtMarkaInfo.Text)
287             {
288                 cmbBoxModelEdit.Items.Add(txtModelInfo.Text);
289                 cmbBoxModelEdit.SelectedIndex = 0;
290             }
291
292
293             foreach (var model in modeller) // modeller listesindeki veriyi combobox'a ekler
294             {
295                 if (cmbBoxModelEdit.Items.Count != 0)
296                 {
297                     if (cmbBoxModelEdit.Items[0].ToString() != model.ToString())
298                     {
299                         cmbBoxModelEdit.Items.Add(model);
300                     }
301                 }
302                 else
303                 {
304                     cmbBoxModelEdit.Items.Add(model);
305                 }
306             }
307
308             reader.Close(); // reader kapat
309
310         }
311         catch (Exception ex)
312         {
313
314             MessageBox.Show(ex.Message);
315         }
316         finally { baglanti.Close(); } // bağlantıyi kapat
317     }
318 }
```

editModeOn() ve editModeOff() methodları içerisindeki kodlar,

```
321  1 reference
322  private void editModeOn() // edit modu açıldığında çalıştırılmak için
323  {
324      lblSegmentEdit.Visible = true;
325      lblGunlukUcret.Visible = false;
326      lblHaftalıkUcret.Visible = false;
327      lblAylıkUcret.Visible = false;
328      lblDurumuInfo.Visible = false;
329      lblDurumuEdit.Visible = true;
330      cmbBoxSegmentEdit.Visible = true;
331      cmbBoxDurumuEdit.Visible = true;
332      txtDurumuInfo.Visible = false;
333      txtGunlukUcret.Visible = false;
334      txtHaftalıkUcret.Visible = false;
335      txtAylıkUcret.Visible = false;
336      txtMarkaInfo.Visible = false;
337      cmbBoxMarkaEdit.Visible = true;
338      txtModelInfo.Visible = false;
339      cmbBoxModelEdit.Visible = true;
340      cmbBoxVitesTuruEdit.Visible = true;
341      cmbBoxYakitTuruEdit.Visible = true;
342  }
343  2 references
344  private void editModeOff() // edit modu kapatıldığında çalıştırılmak için
345  {
346      lblSegmentEdit.Visible = false;
347      lblGunlukUcret.Visible = true;
348      lblHaftalıkUcret.Visible = true;
349      lblAylıkUcret.Visible = true;
350      lblDurumuInfo.Visible = true;
351      lblDurumuEdit.Visible = false;
352      cmbBoxSegmentEdit.Visible = false;
353      cmbBoxDurumuEdit.Visible = false;
354      txtDurumuInfo.Visible = true;
355      txtGunlukUcret.Visible = true;
356      txtHaftalıkUcret.Visible = true;
357      txtAylıkUcret.Visible = true;
358      txtMarkaInfo.Visible = true;
359      txtModelInfo.Visible = true;
360      cmbBoxModelEdit.Visible = false;
361      cmbBoxVitesTuruEdit.Visible = false;
362      cmbBoxYakitTuruEdit.Visible = false;
363  }
```

setEditComboBox() methodu içerisindeki kodlar,

```
365  // set cmbBoxYakitTuruEdit - cmbBoxVitesTuruEdit
366  1 reference
367  private void setEditComboBox()
368  {
369      // comboboxları temizle
370      cmbBoxYakitTuruEdit.Items.Clear();
371      cmbBoxVitesTuruEdit.Items.Clear();
372
373      // varsayılan değerlerin eklenmesi
374      cmbBoxYakitTuruEdit.Items.Add(txtYakitTuruInfo.Text);
375      cmbBoxYakitTuruEdit.SelectedIndex = 0;
376
377      cmbBoxVitesTuruEdit.Items.Add(txtVitesTuruInfo.Text);
378      cmbBoxVitesTuruEdit.SelectedIndex = 0;
379
380      foreach(var item in cmbBoxYakitTuru.Items) // yakıt türlerinin eklenmesi
381      {
382          if(item.ToString() != "Seçiniz..." && item.ToString() != txtYakitTuruInfo.Text)
383          {
384              cmbBoxYakitTuruEdit.Items.Add(item);
385          }
386      }
387      foreach(var item in cmbBoxVitesTuru.Items) // vites türlerinin eklenmesi
388      {
389          if(item.ToString() != "Seçiniz..." && item.ToString() != txtVitesTuruInfo.Text)
390          {
391              cmbBoxVitesTuruEdit.Items.Add(item);
392          }
393      }
394  }
395
396  }
```

FormClosing eventi içerisindeki kodlar,

```
410 1 reference
411  private void VehicleManagementScreen_FormClosing(object sender, FormClosingEventArgs e)
412  {
413      if (e.CloseReason == CloseReason.UserClosing) // Form, kullanıcı tarafından kapatıldıysa
414      {
415          DialogResult result = MessageBox.Show("Uygulamadan çıkmak istediğiniz emin misiniz?", "Uyarı", MessageBoxButtons.YesNo);
416
417          if (result == DialogResult.Yes)
418          {
419              e.Cancel = false;
420
421              if (this.DialogResult != DialogResult.OK) // Eğer form dialog olarak kullanılmiyorsa
422              {
423                  // Login ekranına yönlendir
424                  LoginScreen loginScreen = new LoginScreen();
425                  loginScreen.Show();
426              }
427          }
428          else
429          {
430              e.Cancel = true; // Formun kapatılmasını engelle
431          }
432      }
433      else // Diğer kapatılma sebeplerinde uygulamayı kapat
434      {
435          Application.Exit();
436      }
437 }
```

Textboxların alabileceği değerlerin kontrollerini sağlayan kod blokları,

```
446 // textboxların alabileceği değerlerin belirlenmesi
447 1 reference
448  private void txtYilInfo_KeyPress(object sender, KeyPressEventArgs e)
449  {
450      e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar); // Yalnızca sayısal değerlere izin ver
451  }
452 1 reference
453  private void txtVitesTuruInfo_KeyPress(object sender, KeyPressEventArgs e)
454  {
455      e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
456      && !char.IsSeparator(e.KeyChar); // sadece harf girişi
457  }
458 1 reference
459  private void txtYakitTuruInfo_KeyPress(object sender, KeyPressEventArgs e)
460  {
461      e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
462      && !char.IsSeparator(e.KeyChar); // sadece harf girişi
463  }
464 1 reference
465  private void txtGunlukUcret_KeyPress(object sender, KeyPressEventArgs e)
466  {
467      // Sadece 0-9 arasındaki sayılar ve virgül izin verilsin
468      if (!char.IsDigit(e.KeyChar) && e.KeyChar != '\b')
469      {
470          e.Handled = true; // Karakteri reddet
471      }
472
473      // Sadece bir tane virgül karakteri olmasına izin verilsin
474      /* if (e.KeyChar == ',')
475      {
476          if (txtGunlukUcret.Text.Contains(","))
477          {
478              e.Handled = true; // Karakteri reddet
479          }
480      }*/
481  }
```

“Aracı Düzenle” butonuna tıklandığında çalışacak kod bloğu,

```
1 reference
482 private void btnAracDuzenle_Click(object sender, EventArgs e)
483 {
484     setEnableTextbox(); // textboxları veri girişine aç
485
486     btnAracDuzenle.Visible = false;
487     btnKaydet.Visible = true;
488     dataAraclar.Enabled = false; // gridview dondur
489     btnIptalEt.Visible = true;
490
491     editModeOn(); // Edit için gerekli olan nesneleri gösterir lazım olmayanları gizler.
492     DataGridViewRow selectedRow = dataAraclar.SelectedRows[0];
493
494     //OLD DATA FOR EDIT (eski verileri listeye ekleme)
495     List<String> oldData = new List<string>();
496     oldData.Add(txtPlakaInfo.Text);
497     oldData.Add(txtMarkaInfo.Text);
498     oldData.Add(txtModelInfo.Text);
499     oldData.Add(txtYilInfo.Text);
500     oldData.Add(txtVitesTurInfo.Text);
501     oldData.Add(txtYakitTurInfo.Text);
502     oldData.Add(selectedRow.Cells[6].Value.ToString());
503     oldData.Add(txtDurumuInfo.Text);
504
505     oldPlaka = txtPlakaInfo.Text;
506
507     // Marka Edit
508     cmbBoxMarkaEdit.Items.Clear();
509     cmbBoxMarkaEdit.Items.Add(oldData[1]);
510     cmbBoxMarkaEdit.SelectedIndex = 0;
511     foreach (var item in cmbBoxMarka.Items)
512     {
513
514         if (item.ToString() != "Seçiniz..." && item.ToString() != cmbBoxMarkaEdit.Items[0].ToString())
515         {
516             cmbBoxMarkaEdit.Items.Add(item); // combobox'a markalar ekleniyor
517         }
518     }
519
520     // VitesTuru YakitTuru
521     setEditComboBox();
522
523     // Segment
524     cmbBoxSegmentEdit.Items.Clear();
525     cmbBoxSegmentEdit.Items.Add(oldData[6]);
526     cmbBoxSegmentEdit.SelectedIndex = 0;
527
528     foreach (var item in cmbBoxSegment.Items)
529     {
530         if (item.ToString() != "Seçiniz..." && item.ToString() != oldData[6])
531         {
532             cmbBoxSegmentEdit.Items.Add(item); // segmentler ekleniyor..
533         }
534     }
535
536     //Durumu
537     cmbBoxDurumuEdit.Items.Clear();
538     //cmbBoxDurumuEdit.Items.Add(oldData[7]);
539     switch (oldData[7])
540     {
541         case "True": // seçili satırındaki durum true ise
542         {
543             cmbBoxDurumuEdit.Items.Add("Aktif");
544             break;
545         }
546         default: // false ise
547         {
548             cmbBoxDurumuEdit.Items.Add("Pasif");
549             break;
550         }
551     }
552     cmbBoxDurumuEdit.SelectedIndex = 0;
553
554     foreach (var item in cmbBoxDurum.Items)
555     {
556         if (item.ToString() != "Seçiniz..." && item.ToString() != cmbBoxDurumuEdit.Items[0].ToString())
557         {
558             cmbBoxDurumuEdit.Items.Add(item); // durum ekleniyor
559         }
560     }
561 }
```

updateData() methodu içerisindeki kodlar,

```
1 reference
563 private void updateData(List<String> paramListe) // veritabanı update işlemi
564 {
565     try
566     {
567         if (baglanti.State == ConnectionState.Closed)
568         {
569             baglanti.Open();
570         }
571         // PLAKAYA AİT ARAÇ ID ALMA
572         SqlCommand cmdAracId = new SqlCommand("getAracId", baglanti); // sqlCommand oluşturma
573         cmdAracId.CommandType = CommandType.StoredProcedure; // command türünü storedProcedure olarak ayarlama
574         SqlParameter paramPlaka = new SqlParameter("@paramPlaka", oldPlaka); // parametre oluşturma
575         paramPlaka.Direction = ParameterDirection.Input; // parametre türünü input olarak belirleme
576         cmdAracId.Parameters.Add(paramPlaka); // parametre geçme
577
578         SqlParameter paramAracId = new SqlParameter("@Id", SqlDbType.Int);
579         paramAracId.Direction = ParameterDirection.Output; // parametre türünü output ayarlama
580         cmdAracId.Parameters.Add(paramAracId); // parametre geçiliyor
581
582         // SqlCommand nesnesini çalıştırma
583         cmdAracId.ExecuteNonQuery();
584
585         // Çıktı parametresindeki değeri alma
586         int aracId = (int)cmdAracId.Parameters["@Id"].Value;      // ARAÇ ID
587         // MARKA ID ALMA
588         SqlCommand cmd = new SqlCommand("getMarkaId", baglanti); // sqlCommand oluşturma
589         cmd.CommandType = CommandType.StoredProcedure; // command türünü storedProcedure olarak belirleme
590         SqlParameter paramMarkaAdi = new SqlParameter("@paramMarkaAdi", paramListe[1]); // parametre oluşturma
591         paramMarkaAdi.Direction = ParameterDirection.Input; // parametre türünü input olarak ayarlar
592         cmd.Parameters.Add(paramMarkaAdi); // parametre geçme
593
594         SqlParameter paramId = new SqlParameter("@Id", SqlDbType.Int); // parametre oluşturma
595         paramId.Direction = ParameterDirection.Output; // parametre türünü output olarak ayarlar
596         cmd.Parameters.Add(paramId); // parametre geçme
597
598         // SqlCommand nesnesini çalıştırma
599         cmd.ExecuteNonQuery();
600
601         // Çıktı parametresindeki değeri alma
602         int markaId = (int)cmd.Parameters["@Id"].Value;      // MARKA ID
603
604
605         // MODEL ID ALMA
606         SqlCommand cmdModelId = new SqlCommand("getModelId", baglanti); // sqlCommand oluşturma
607         cmdModelId.CommandType = CommandType.StoredProcedure; // command türünü belirleme
608         SqlParameter paramModelAdi = new SqlParameter("@paramModelAdi", paramListe[2]); // parametre oluşturma
609         paramModelAdi.Direction = ParameterDirection.Input; // parametre türünü input olarak ayarlar
610         cmdModelId.Parameters.Add(paramModelAdi); // parametre geçme
611
612         SqlParameter paramModelId = new SqlParameter("@Id", SqlDbType.Int); //parametre oluşturma
613         paramModelId.Direction = ParameterDirection.Output; // parametre türünü output olarak belirleme
614         cmdModelId.Parameters.Add(paramModelId); // parametre geçme
615
616         // SEGMENT ID ALMA
617
618         SqlCommand cmdSegmentId = new SqlCommand("getSegmentId", baglanti); // sqlCommand oluşturma
619         cmdSegmentId.CommandType = CommandType.StoredProcedure; // command türü belirleme
620         SqlParameter paramSegmentAdi = new SqlParameter("@paramSegmentAdi", paramListe[6]); // parametre oluşturma
621         paramSegmentAdi.Direction = ParameterDirection.Input; // parametre türünü input yapar
622         cmdSegmentId.Parameters.Add(paramSegmentAdi); // parametre geçme
623
624         SqlParameter paramSegmentId = new SqlParameter("@Id", SqlDbType.Int); // parametre oluşturma
625         paramSegmentId.Direction = ParameterDirection.Output; // parametre türünü output olarak ayarlama
626         cmdSegmentId.Parameters.Add(paramSegmentId); // parametre geçme
627
628         // SqlCommand nesnesini çalıştırma
629         cmdSegmentId.ExecuteNonQuery();
630
631         // Çıktı parametresindeki değeri alma
632         int segmentId = (int)cmdSegmentId.Parameters["@Id"].Value;      //SEGMENT ID
633         // *** SQL UPDATE ***
634         string query = "update tblArac SET Plaka=@plaka, MarkaId=@markaId, " +
635             "ModelId = @modelId , " +
636             "Yili = @yili , " +
637             "VitesTuru = @vitesTuru , " +
638             "YakitTuru = @yakitTuru , " +
639             "SegmentId = @segmentId , " +
640             "Durumu = @durumu where Id = @Id "; // where koşulu !!!
641
642
643
644
645
646
647
648
649
650
651
652
```

```

653     SqlCommand cmdUpdate = new SqlCommand(query,baglanti); // sqlCommand oluşturma
654     // parametre geçiriliyor...
655     cmdUpdate.Parameters.AddWithValue("@plaka", paramliste[0]);
656     cmdUpdate.Parameters.AddWithValue("@markaId", markaId);
657     cmdUpdate.Parameters.AddWithValue("@modelId", modelId);
658     cmdUpdate.Parameters.AddWithValue("@yili", paramliste[3]);
659     cmdUpdate.Parameters.AddWithValue("@vitesTuru", paramliste[4]);
660     cmdUpdate.Parameters.AddWithValue("@yakitTuru", paramliste[5]);
661     cmdUpdate.Parameters.AddWithValue("@segmentId", segmentId);
662     cmdUpdate.Parameters.AddWithValue("@durumu", paramliste[7]);
663     cmdUpdate.Parameters.AddWithValue("@Id", araciD);
664
665     cmdUpdate.ExecuteNonQuery(); //execute
666
667     MessageBox.Show("Araç Başarıyla Güncellendi.");
668     getAllCarsData(); // datagridview refresh
669
670
671 }
672 catch (Exception ex)
673 {
674
675     MessageBox.Show(ex.Message);
676 }
677 finally
678 {
680     baglanti.Close(); // bağlantıyi kapat
681 }
682
683
684 }

```

“Kaydet” butonuna tıklandığında çalışacak kod bloğu,

```

687     private void btnKaydet_Click(object sender, EventArgs e)
688     {
689
690
691         if(errorProvider.GetError(txtPlakaInfo) == "" && errorProvider.GetError(txtYilInfo) == "") // aktif errorProvider(hata) yok ise
692         {
693             setDisableTextbox();
694             dataAraclar.ReadOnly = true; // datagridviewi ReadOnly olarak ayarlama
695             btnKaydet.Visible = false; // kaydet butonunu gizle
696             btnAracDuzenle.Visible = true; //butonu aktif et
697             dataAraclar.Enabled = true; // datagridviewi aktif et
698             btnIptalEt.Visible = false; // iptal et butonunu gizler
699             editModeOff(); // Edit için gerekli olan nesneleri gizler
700
701             // Yeni veriler (güncelleme için)
702             List<String> newData = new List<string>();
703             newData.Add(txtPlakaInfo.Text);
704             newData.Add(cmbBoxMarkaEdit.SelectedItem.ToString());
705             newData.Add(cmbBoxModelEdit.SelectedItem.ToString());
706             newData.Add(txtYilInfo.Text);
707             newData.Add(cmbBoxVitesTuruEdit.SelectedItem.ToString());
708             newData.Add(cmbBoxYakitTuruEdit.SelectedItem.ToString());
709             newData.Add(cmbBoxSegmentEdit.SelectedItem.ToString());
710             newData.Add(cmbBoxDurumuEdit.SelectedItem.ToString());
711             switch (cmbBoxDurumuEdit.SelectedItem.ToString())
712             {
713                 case "Aktif":
714                 {
715                     newData.Add("true");
716                     break;
717                 }
718
719                 default:
720                 {
721                     newData.Add("false");
722                     break;
723                 }
724             }
725             updateData(newData); //sql güncelleştirme
726         }
727         else
728         {
729             MessageBox.Show("Lütfen hataları düzeltin.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
730         }

```

“Araç Ekle” butonuna tıklandığında çalışacak kod bloğu,

```
1 reference
752 private void btnAracEkle_Click(object sender, EventArgs e)
753 {
754     // Veri Kontrolü
755     // Tüm hataların düzeltilmesi kontrol ediliyor.
756     if (
757         errorProvider.GetError(txtAddPlaka) == ""
758         && errorProvider.GetError(txtAddYil) == ""
759         && errorProvider.GetError(cmbBoxMarka) == ""
760         && errorProvider.GetError(cmbBoxModel) == ""
761         && errorProvider.GetError(cmbBoxSegment) == ""
762         && errorProvider.GetError(cmbBoxVitesTuru) == ""
763         && errorProvider.GetError(cmbBoxYakitTuru) == ""
764         && errorProvider.GetError(cmbBoxDurum) == ""))
765     {
766         // Textboxların içerisinde herhangi bir veri girişi olmadığından
767         if (
768             txtAddPlaka.Text == "Plaka"
769             || txtAddYil.Text == "Yıl"
770             || cmbBoxDurum.SelectedItem.ToString() == "Seçiniz..."
771             || cmbBoxMarka.SelectedItem.ToString() == "Seçiniz..."
772             || cmbBoxModel.SelectedItem.ToString() == "Seçiniz..."
773             || cmbBoxYakitTuru.SelectedItem.ToString() == "Seçiniz..."
774             || cmbBoxVitesTuru.SelectedItem.ToString() == "Seçiniz..."
775             || cmbBoxSegment.SelectedItem.ToString() == "Seçiniz...")
776         {
777             MessageBox.Show("Lütfen tüm bilgileri eksiksiz doldurun.");
778         }
779     }
780     else
781     {
782
783
784
785         if (baglanti.State == ConnectionState.Closed)
786         { baglanti.Open(); }
787
788         try
789         {
790             string query = "select * from tblArac where Plaka = @plaka "; // sql sorgu cümlesi oluşturma
791             SqlCommand cmd = new SqlCommand(query, baglanti); // sqlCommand oluşturma
792             cmd.Parameters.AddWithValue("@plaka", txtAddPlaka.Text); // parametre geçme
793             SqlDataReader reader = cmd.ExecuteReader(); // execute işlemi ve dönen verinin reader da tutulması
794             Boolean hasRows = reader.HasRows; // reader da veri var mı
795             reader.Close(); // reader kapat
796             if (hasRows) // true ise
797             {
798                 MessageBox.Show("Girdiğiniz plaka ait kayıt bulundu !");
799             }
800             else
801             {
802                 string durum = cmbBoxDurum.SelectedItem.ToString();
803                 Boolean durumu = false;
804                 if (durum == "Aktif")
805                 {
806                     durumu = true;
807                 }
808                 SqlCommand cmdAdd = new SqlCommand("addCar", baglanti); // sqlCommand oluşturma
809                 cmdAdd.CommandType = CommandType.StoredProcedure; // command türü belirleme
810                 // *** parametre geçiriliyor ***
811                 cmdAdd.Parameters.AddWithValue("@paramPlaka", txtAddPlaka.Text.ToUpper());
812                 cmdAdd.Parameters.AddWithValue("@paramMarka", cmbBoxMarka.SelectedItem.ToString());
813                 cmdAdd.Parameters.AddWithValue("@paramModel", cmbBoxModel.SelectedItem.ToString());
814                 cmdAdd.Parameters.AddWithValue("@paramYil", txtAddYil.Text);
815                 cmdAdd.Parameters.AddWithValue("@paramVitesTuru", cmbBoxVitesTuru.SelectedItem.ToString());
816                 cmdAdd.Parameters.AddWithValue("@paramYakitTuru", cmbBoxYakitTuru.SelectedItem.ToString());
817                 cmdAdd.Parameters.AddWithValue("@paramSegment", cmbBoxSegment.SelectedItem.ToString());
818                 cmdAdd.Parameters.AddWithValue("@paramDurumu", durumu);
819
820                 cmdAdd.ExecuteNonQuery(); // execute
821                 MessageBox.Show(txtAddPlaka.Text + " Plakalı " + cmbBoxMarka.SelectedItem.ToString() + " marka araç başarıyla eklendi.
822                 getAllCarsData(); // datagridview refresh
823                 defaultValuesAracEkle(); // default değerlerin getirilmesi
824             }
825
826         }
827         catch (Exception ex)
828         {
829             MessageBox.Show(ex.Message);
830         }
831         finally { baglanti.Close(); } // bağlantıyi kapat
832
833     }
834 }
835 }
836 else
837 {
838     MessageBox.Show("Lütfen hataları düzeltin.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
839 }
840 }
```

defaultValuesAracEkle() methodu içerisindeki kodlar,

```
1 reference  
842     private void defaultValuesAracEkle() //araç ekleme işleminden sonra default değerlerin getirilmesi için  
843     {  
844         txtAddPlaka.Text = "Plaka";  
845         txtAddYil.Text = "Yıl";  
846         cmbBoxDurum.SelectedIndex = 0;  
847         cmbBoxMarka.SelectedIndex = 0;  
848         cmbBoxModel.SelectedIndex = 0;  
849         cmbBoxSegment.SelectedIndex = 0;  
850         cmbBoxVitesTuru.SelectedIndex = 0;  
851         cmbBoxYakitTuru.SelectedIndex = 0;  
852     }  
853
```

cmbBoxMarka ve cmbBoxMarkaEdit nesnelerinin SelectedIndexChanged eventi içerisindeki kodlar,

```
1 reference  
854     private void cmbBoxMarka_SelectedIndexChanged(object sender, EventArgs e)  
855     {  
856         if(cmbBoxMarka.SelectedIndex != 0) // marka seçildiğinde o markaya ait modelleri getir  
857         {  
858             getModel();  
859         }  
860     }  
861  
862     1 reference  
863     private void cmbBoxMarkaEdit_SelectedIndexChanged(object sender, EventArgs e)  
864     {  
865         getModelForEdit(); // güncelleme işlemi için marka seçildiğinde seçilen markanın modellerini getirir  
866     }  
867
```

“İptal Et” butonuna tıklandığında çalışacak kod bloğu,

```
1 reference  
868     private void btnIptalEt_Click(object sender, EventArgs e) // iptal et butonuna tıklandığında  
869     {  
870         setDisableTextbox();  
871         dataAraclar.ReadOnly = true;  
872         btnKaydet.Visible = false;  
873         btnAracDuzenle.Visible = true;  
874         dataAraclar.Enabled = true; //datagridviewi aktif eder  
875         editModeOff(); // Edit için gerekli olan nesneleri gizler.  
876         btnIptalEt.Visible = false;  
877         errorProvider.SetError(txtPlakaInfo,""); // hata varsa kaldırır  
878         errorProvider.SetError(txtYilInfo, ""); // hata varsa kaldırır  
879         getSelectedCarInfo(); // seçili araç bilgilerini getir  
880     }  
881
```

“Aracı Sil” butonuna tıklandığında çalışacak kod bloğu,

```
1 reference  
882     private void btnAraciSil_Click(object sender, EventArgs e) // araç silme işlemi  
883     {  
884         // onay mesajı  
885         DialogResult result = MessageBox.Show(  
886             txtPlakaInfo.Text + " Plakalı aracı silmek istediğiniz emin misiniz? Bu işlem geri alınamaz.", "Uyarı", MessageBoxButtons.YesNo);  
887  
888         if (result == DialogResult.Yes) // onay verildiyse  
889         {  
890             try  
891             {  
892                 if (baglanti.State == ConnectionState.Closed)  
893                 {  
894                     baglanti.Open();  
895                 }  
896                 // PLAKAYA AİT ARAÇ ID ALMA  
897                 SqlCommand cmdAracId = new SqlCommand("getAracId", baglanti);  
898                 cmdAracId.CommandType = CommandType.StoredProcedure; //command türü belirleme  
899                 string plaka = txtPlakaInfo.Text; // id si almak istenilen plaka  
900                 SqlParameter paramPlaka = new SqlParameter("@paramPlaka", plaka); // parametre oluşturma  
901                 paramPlaka.Direction = ParameterDirection.Input; // parametre türünü input olarak ayarlar  
902                 cmdAracId.Parameters.Add(paramPlaka); // parametre geçme  
903  
904                 SqlParameter paramAracId = new SqlParameter("@Id", SqlDbType.Int); // parametre oluşturma  
905                 paramAracId.Direction = ParameterDirection.Output; // parametre türünü output olarak belirler  
906                 cmdAracId.Parameters.Add(paramAracId); //parametre geçme  
907  
908                 // SqlCommand nesnesini çalıştırma  
909                 cmdAracId.ExecuteNonQuery();  
910             }  
911         }  
912     }  
913
```

```

911 // Çıktı parametresindeki değeri alma
912 int aracId = (int)cmdAracId.Parameters["@Id"].Value;      // ARAÇ ID
913
914 // ARACI SİL
915 string query = "delete from tblArac where Id = @Id"; // sql sorgu cümlesi
916 SqlCommand cmd = new SqlCommand(query, baglanti); // sqlCommand oluşturma
917 cmd.Parameters.AddWithValue("@Id", aracId); // parametre geçme
918 cmd.ExecuteNonQuery(); // execute işlemi
919 MessageBox.Show(plaka + " Plakalı araç silindi.");
920 getAllCarsData(); // refresh datagridview
921 lblAracYonetim.Focus(); // focusu textlerden alır
922 }
923 catch (Exception ex)
924 {
925     MessageBox.Show(ex.Message);
926 }
927 finally
928 {
929     baglanti.Close(); // bağlantıyi kapat
930 }
931 }
932 }
```

Textbox placeholder kodları,

```

936 // *** Textbox Placeholder Kodları ***
937 1 reference
938 private void txtAddPlaka_Enter(object sender, EventArgs e)
939 {
940     if (txtAddPlaka.Text == "Plaka")
941     {
942         txtAddPlaka.Text = "";
943         txtAddPlaka.ForeColor = Color.Black;
944     }
945 }
946 1 reference
947 private void txtAddPlaka_Leave(object sender, EventArgs e)
948 {
949     if (txtAddPlaka.Text == "")
950     {
951         txtAddPlaka.Text = "Plaka";
952         txtAddPlaka.ForeColor = Color.Gray;
953     }
954 }
955 1 reference
956 private void txtAddYil_Enter(object sender, EventArgs e)
957 {
958     if (txtAddYil.Text == "Yıl")
959     {
960         txtAddYil.Text = "";
961         txtAddYil.ForeColor = Color.Black;
962     }
963 }
964 1 reference
965 private void txtAddYil_Leave(object sender, EventArgs e)
966 {
967     if (txtAddYil.Text == "")
968     {
969         txtAddYil.Text = "Yıl";
970         txtAddYil.ForeColor = Color.Gray;
971     }
972 }
```

Textbox ve ComboBoxların validating eventi içerisindeki kodlar,

```
973 // *** Validation ***
974 1 reference
975 private void txtAddPlaka_Validating(object sender, CancelEventArgs e)
976 {
977     if (string.IsNullOrWhiteSpace(txtAddPlaka.Text) || txtAddPlaka.Text == "Plaka")
978     {
979         errorProvider.SetError(txtAddPlaka, "Lütfen plakayı girin.");
980     }
981     else
982     {
983         errorProvider.SetError(txtAddPlaka, null);
984     }
985 }
986 1 reference
987 private void cmbBoxMarka_Validating(object sender, CancelEventArgs e)
988 {
989     if (cmbBoxMarka.SelectedIndex == 0)
990     {
991         errorProvider.SetError(cmbBoxMarka, "Lütfen eklemek istediğiniz aracın markasını seçin.");
992     }
993     else
994     {
995         errorProvider.SetError(cmbBoxMarka, null);
996     }
997 }
998 1 reference
999 private void cmbBoxModel_Validating(object sender, CancelEventArgs e)
1000 {
1001     if (cmbBoxModel.SelectedIndex == 0)
1002     {
1003         errorProvider.SetError(cmbBoxModel, "Lütfen eklemek istediğiniz aracın modelini seçin.");
1004     }
1005     else
1006     {
1007         errorProvider.SetError(cmbBoxModel, null);
1008     }
1009 }
1010 1 reference
1011 private void txtAddYil_Validating(object sender, CancelEventArgs e)
1012 {
1013     if (string.IsNullOrWhiteSpace(txtAddYil.Text) || txtAddYil.Text == "Yıl")
1014     {
1015         errorProvider.SetError(txtAddYil, "Lütfen aracın yılını girin.");
1016     }
1017     else
1018     {
1019         errorProvider.SetError(txtAddYil, null);
1020     }
1021 }
1022 1 reference
1023 private void cmbBoxVitesTuru_Validating(object sender, CancelEventArgs e)
1024 {
1025     if (cmbBoxVitesTuru.SelectedIndex == 0)
1026     {
1027         errorProvider.SetError(cmbBoxVitesTuru, "Lütfen eklemek istediğiniz vites türünü seçin.");
1028     }
1029     else
1030     {
1031         errorProvider.SetError(cmbBoxVitesTuru, null);
1032     }
1033 }
1034 1 reference
1035 private void cmbBoxYakitTuru_Validating(object sender, CancelEventArgs e)
1036 {
1037     if (cmbBoxYakitTuru.SelectedIndex == 0)
1038     {
1039         errorProvider.SetError(cmbBoxYakitTuru, "Lütfen eklemek istediğiniz aracın yakıt türünü seçin.");
1040     }
1041     else
1042     {
1043         errorProvider.SetError(cmbBoxYakitTuru, null);
1044     }
1045 }
1046 1 reference
1047 private void cmbBoxSegment_Validating(object sender, CancelEventArgs e)
1048 {
1049     if (cmbBoxSegment.SelectedIndex == 0)
1050     {
1051         errorProvider.SetError(cmbBoxSegment, "Lütfen eklemek istediğiniz aracın segmentini seçin.");
1052     }
1053     else
1054     {
1055         errorProvider.SetError(cmbBoxSegment, null);
1056     }
1057 }
1058 1 reference
1059 private void cmbBoxDurum_Validating(object sender, CancelEventArgs e)
1060 {
1061     if (cmbBoxDurum.SelectedIndex == 0)
1062     {
1063         errorProvider.SetError(cmbBoxDurum, "Lütfen eklemek istediğiniz aracın görüntülenme durumunu seçin.");
1064     }
1065     else
1066     {
1067         errorProvider.SetError(cmbBoxDurum, null);
1068     }
1069 }
```

```

1 reference
1070 private void txtPlakaInfo_Validating(object sender, CancelEventArgs e)
1071 {
1072     if (string.IsNullOrEmpty(txtPlakaInfo.Text))
1073     {
1074         errorProvider.SetError(txtPlakaInfo, "Lütfen aracın plakasını girin.");
1075     }
1076     else
1077     {
1078         errorProvider.SetError(txtPlakaInfo, null);
1079     }
1080 }
1081
1 reference
1082 private void txtYilInfo_Validating(object sender, CancelEventArgs e)
1083 {
1084     if (string.IsNullOrEmpty(txtYilInfo.Text))
1085     {
1086         errorProvider.SetError(txtYilInfo, "Lütfen aracın plakasını girin.");
1087     }
1088     else
1089     {
1090         errorProvider.SetError(txtYilInfo, null);
1091     }
1092 }

```

Menü butonlarına tıklandığında çalışacak kod blokları,

```

1094 // Menü butonları
1095
1096 private void LnkLabelAracKiralamaIslemleri_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e) //araç kiralama ekranına geçiş
1097 {
1098     this.Hide();
1099     MainScreen mainScreen = new MainScreen();
1100     mainScreen.Show();
1101 }
1102
1103 private void LnkLabelAracTeslim_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e) // araç teslim ekranına geçiş
1104 {
1105     this.Hide();
1106     VehicleDeliveryScreen vehicleDeliveryScreen = new VehicleDeliveryScreen();
1107     vehicleDeliveryScreen.Show();
1108 }
1109
1110 private void LnkLabelCikis_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e) // çıkış yap butonu
1111 {
1112     //onay mesajı
1113     DialogResult result = MessageBox.Show("Hesabınızdan çıkmak istediğinizden emin misiniz?", "Uyarı", MessageBoxButtons.YesNo);
1114
1115     if (result == DialogResult.Yes)
1116     {
1117
1118         if (this.DialogResult != DialogResult.OK) // Eğer form dialog olarak kullanılmıyorsa
1119         {
1120             // Login ekranına yönlendir
1121             LoginScreen loginScreen = new LoginScreen();
1122             loginScreen.Show();
1123             this.Hide();
1124         }
1125     }
1126 }
1127
1128 private void LnkLabelMusteriYonetim_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e) // müşteri yönetim ekranına geçiş
1129 {
1130     CustomerManagementScreen customerManagementScreen = new CustomerManagementScreen();
1131     customerManagementScreen.Show();
1132     this.Hide();
1133 }
1134
1135 private void LnkLabelHesapAyar_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e) //hesap ayarları ekranına geçiş
1136 {
1137     AccountSettingsScreen accountSettingsScreen = new AccountSettingsScreen();
1138     accountSettingsScreen.Show();
1139     this.Hide();
1140 }

```

E7-Müşteri Yönetim Ekranı

ID	TC	Ehliyet No	Müşteri Adı	Müşteri Soyadı	E-posta	Telefon No	Durumu
1	22255566644	A55123	Emre	Can	emre@gmail.com	5078976675	True
2	42152342152	A41231	Ahmet	Gün	ahmet@gmail.com	5554441123	True
3	42112332445	A41264	Mehmet	Öz	mehmet@gmail.com	1423456365	False
12	44444444444	A31232	deneme	Deadsa	dsadsa@gmail.com	4444444444	True

Kodlar1

Veritabanı bağlantısı , ErrorProvider nesnesinin bağlanması ve form çağrıldığında çalışacak methodlar:

```
18 //Veritabanı bağlantı işlemi
19 SqlConnection bağlantı = new SqlConnection("Data Source=DESKTOP-BTIS91A;Initial Catalog=Db_AracKiralama;Integrated Security=True");
20 //ErrorProvider nesnesinin çağrılması
21 ErrorProvider errorProvider = new ErrorProvider();
22
23
24     4 references
25 public CustomerManagementScreen()
26 {
27     InitializeComponent();
28
29     // Karşılama Mesajı
30     lblLeftName.Text = "Merhaba , " + LoginScreen.userName;
31     cmbBoxDurumu.SelectedIndex = 0; // combobox in default değerinin seçilmesi
32     getAllCustomersData();
```

getSelectedCustomerInfo() methodu içerisindeki kodlar,

```
60     2 references
61     private void getSelectedCustomerInfo() // seçili müşteri bilgilerini ilgili textboxlara yazdırır
62     {
63         if (dataMusteri.SelectedRows.Count > 0) // seçili satır var ise
64         {
65             DataGridViewRow selectedRow = dataMusteri.SelectedRows[0];
66             if (selectedRow.Cells[0].Value != null)
67             {
68                 txtIdInfo.Text = selectedRow.Cells[0].Value.ToString();
69                 txtTcInfo.Text = selectedRow.Cells[1].Value.ToString();
70                 txtEhliyetInfo.Text = selectedRow.Cells[2].Value.ToString();
71                 txtMusteriAdiInfo.Text = selectedRow.Cells[3].Value.ToString();
72                 txtMusteriSoyadiInfo.Text = selectedRow.Cells[4].Value.ToString();
73                 txtEpostaInfo.Text = selectedRow.Cells[5].Value.ToString();
74                 txtTelefonNoInfo.Text = selectedRow.Cells[6].Value.ToString();
75                 txtDurumuInfo.Text = selectedRow.Cells[7].Value.ToString();
76             }
77         }
78     }
```

getAllCustomersData() methodu çağrıldığında çalışacak kod bloğu,

```
36     4 references
37     private void getAllCustomersData()
38     {
39         if (baglanti.State == ConnectionState.Open)
40         { baglanti.Close(); }
41
42         DataTable dataTable = new DataTable(); // DataTable nesnesi oluşturma
43         try
44         {
45             baglanti.Open(); // bağlantıyi aç
46             string query = "select * from tblMusteri"; // sql sorgu cümlesi
47             SqlCommand cmd = new SqlCommand(query, baglanti); // sqlCommand oluşturma
48             SqlDataReader reader = cmd.ExecuteReader(); // execute işlemi ve reader nesnesine dönen verinin aktarılması
49             dataTable.Load(reader); // dataTable nesnesine reader içindeki verinin aktarımı
50             reader.Close(); // reader kapat
51             dataMusteri.DataSource = dataTable; // dataTable içerisindeki veriyi datagridview'e ekleme
52         }
53         catch (Exception ex)
54         {
55             MessageBox.Show(ex.Message);
56         }
57         finally { baglanti.Close(); } // bağlantıyi kapat
58     }
```

setEnableTextbox() ve setDisableTextbox() methodları içerisindeki kodlar,

```
80     1 reference
81     private void setEnableTextbox() // ilgili textboxları enable yapar
82     {
83         txtTcInfo.Enabled = true;
84         txtEhliyetInfo.Enabled = true;
85         txtMusteriAdiInfo.Enabled = true;
86         txtMusteriSoyadiInfo.Enabled = true;
87         txtEpostaInfo.Enabled = true;
88         txtTelefonNoInfo.Enabled = true;
89     }
90     2 references
91     private void setDisableTextbox() // ilgili textboxları disable yapar
92     {
93         txtTcInfo.Enabled = false;
94         txtEhliyetInfo.Enabled = false;
95         txtMusteriAdiInfo.Enabled = false;
96         txtMusteriSoyadiInfo.Enabled = false;
97         txtEpostaInfo.Enabled = false;
98         txtTelefonNoInfo.Enabled = false;
99     }
```

IsValidEmail() methodu içerisindeki kodlar,

```
99     2 references
100    private bool IsValidEmail(string email) // Epostanın doğru formatta olup olmadığını kontrol eder true - false döner
101    {
102        if (string.IsNullOrWhiteSpace(email))
103            return false;
104        try
105        {
106            MailAddress mailAddress = new MailAddress(email); // Girilen emaili formata uygun değilse hata fırlatır ve catch bloğu çalışır
107            return true;
108        }
109        catch (FormatException)
110        {
111            return false;
112        }
113    }
```

Datagridview SelectionChanged eventi içerisindeki kodlar,

```
166     1 reference
167     private void dataMusteri_SelectionChanged(object sender, EventArgs e) // datagridviewde seçilen satırın bilgilerini getirir
168     {
169         getSelectedCustomerInfo();
170     }
```

FormClosing eventi içerisindeki kodlar,

```
1 reference
120  private void CustomerManagementScreen_FormClosing(object sender, FormClosingEventArgs e)
121  {
122      if (e.CloseReason == CloseReason.UserClosing) // Form, kullanıcı tarafından kapatıldıysa
123      {
124          DialogResult result = MessageBox.Show("Uygulamadan çıkmak istediğiniz emin misiniz?", "Uyarı", MessageBoxButtons.YesNo);
125
126          if (result == DialogResult.Yes)
127          {
128              e.Cancel = false;
129
130              if (this.DialogResult != DialogResult.OK) // Eğer form dialog olarak kullanılmıyorsa
131              {
132                  // Login ekranına yönlendir
133                  LoginScreen loginScreen = new LoginScreen();
134                  loginScreen.Show();
135              }
136          }
137          else
138          {
139              e.Cancel = true; // Formun kapatılmasını engelle
140          }
141      }
142  else // Diğer kapatılma sebeplerinde uygulamayı kapat
143  {
144      Application.Exit();
145  }
146 }
```

“Müşteriyi Düzenle” butonuna tıklandığında çalışacak kod bloğu,

```
1 reference
155  private void btnMusteriDuzenle_Click(object sender, EventArgs e) // müşteriyi düzenle butonuna basıldığında çalışacak event
156  {
157      btnMusteriDuzenle.Visible = false;
158      btnKaydet.Visible = true;
159      btnIptalEt.Visible = true;
160      setEnableTextBox();
161      dataMusteri.Enabled = false;
162      // dataMusteri.ReadOnly= false;
163      cmbBoxDurumuEdit.Visible = true;
164      txtDurumuInfo.Visible = false;
165
166      //OLD DATA FOR EDIT (eski verileri bir list içerisinde tutmak için)
167      List<String> oldData = new List<string>();
168      oldData.Add(txtIdInfo.Text);
169      oldData.Add(txtTcInfo.Text);
170      oldData.Add(txtEhliyetInfo.Text);
171      oldData.Add(txtMusteriAdiInfo.Text);
172      oldData.Add(txtMusteriSoyadiInfo.Text);
173      oldData.Add(txtEpostaInfo.Text);
174      oldData.Add(txtTelefonNoInfo.Text);
175      oldData.Add(txtDurumuInfo.Text);
176
177      // DURUMU COMBOBOX
178      cmbBoxDurumuEdit.Items.Clear(); // comboboxın içeriği temizlendi.
179      //item ekleme
180      cmbBoxDurumuEdit.Items.Add("Aktif");
181      cmbBoxDurumuEdit.Items.Add("Pasif");
182      switch (oldData[7])
183      {
184          case "True":
185          {
186              cmbBoxDurumuEdit.SelectedIndex = 0;
187              break;
188          }
189          default:
190          {
191              cmbBoxDurumuEdit.SelectedIndex = 1;
192              break;
193          }
194      }
195 }
```

“Kaydet” butonuna tıklandığında çalışacak kod bloğu,

```
1 reference
196 private void btnKaydet_Click(object sender, EventArgs e)
{
197
198
199     //hata yoksa
200     if (errorProvider.GetError(txtTcInfo) == ""
201         && errorProvider.GetError(txtEhliyetInfo) == ""
202         && errorProvider.GetError(txtMusteriAdiInfo) == ""
203         && errorProvider.GetError(txtMusteriSoyadiInfo) == ""
204         && errorProvider.GetError(txtEpostaInfo) == ""
205         && errorProvider.GetError(txtTelefonNoInfo) == "")
206     {
207
208         //dataMusteri.ReadOnly = true;
209         dataMusteri.Enabled = true;
210         btnMusteriDuzenle.Visible = true;
211         btnKaydet.Visible = false;
212         btnIptalEt.Visible = false;
213         setDisableTextbox();
214         cmbBoxDurumuEdit.Visible = false;
215         txtDurumuInfo.Visible = true;
216
217         // Yeni veriler(veritabanına gidecek veriler önce bu listeye eklenip daha sonra gerekli düzenlemeler yapılacak)
218         List<String> newData = new List<string>();
219         newData.Add(txtTcInfo.Text);
220         newData.Add(txtEhliyetInfo.Text);
221         newData.Add(txtMusteriAdiInfo.Text);
222         newData.Add(txtMusteriSoyadiInfo.Text);
223         newData.Add(txtEpostaInfo.Text);
224         newData.Add(txtTelefonNoInfo.Text);
225         switch (cmbBoxDurumuEdit.SelectedItem.ToString())
226         {
227             case "Aktif": //combobox içerisindeKİ Aktif seçilirse veritabanına true olarak gider
228             {
229                 newData.Add("true");
230                 break;
231             }
232
233             default: // aksi takdirde false olarak eklenir
234             {
235                 newData.Add("false");
236                 break;
237             }
238         }
239
240         updateData(newData);    //sql güncelleştirme
241     }
242     else
243     {
244         MessageBox.Show("Lütfen hataları düzeltin.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
245     }
246 }
```

“İptal Et” butonuna tıklandığında çalışacak kodlar,

```
1 reference
248 private void btnIptalEt_Click(object sender, EventArgs e) // iptal et butonuna tıklandığında çalışacak event
{
249
250     btnMusteriDuzenle.Visible = true; // düzenle butonunu görünürlük
251     btnKaydet.Visible = false; // kaydet butonunu gizle
252     lblMusteriYonetim.Focus();
253     btnIptalEt.Visible = false; // iptal et butonunu gizle
254     setDisableTextbox(); // textboxları disable yap
255     dataMusteri.Enabled = true; // datagridviewi aktif et
256     cmbBoxDurumuEdit.Visible = false; // comboboxı gizle
257     txtDurumuInfo.Visible = true; //durumu textboxını göster
258     // hataları kaldır
259     errorProvider.SetError(txtTcInfo, "");
260     errorProvider.SetError(txtEhliyetInfo, "");
261     errorProvider.SetError(txtMusteriAdiInfo, "");
262     errorProvider.SetError(txtMusteriSoyadiInfo, "");
263     errorProvider.SetError(txtEpostaInfo, "");
264     errorProvider.SetError(txtTelefonNoInfo, "");
265     getSelectedCustomerInfo(); // seçili müşteri bilgilerini getir
266 }
```

updateData() methodu içerisindeki kodlar,

```
268     1 reference
269     private void updateData(List<String> paramListe)
270     {
271         try
272         {
273             if (baglanti.State == ConnectionState.Closed)
274             {
275                 baglanti.Open();
276             }
277
278             // *** SQL UPDATE ***
279             string query =
280                 "update tblMusteri SET Tc=@tc, " +
281                 "EhliyetNo=@ehliyetNo, " +
282                 "Ad = @ad , Soyad = @soyad , " +
283                 "Mail = @mail , TelNo = @telNo , " +
284                 "Durumu = @durumu where Id = @Id "; // where koşulu !!!
285             SqlCommand cmdUpdate = new SqlCommand(query, baglanti); //sqlCommand nesnesi oluşturma
286             // parametre geçiriliyor...
287             cmdUpdate.Parameters.AddWithValue("@tc", paramListe[0]);
288             cmdUpdate.Parameters.AddWithValue("@ehliyetNo", paramListe[1]);
289             cmdUpdate.Parameters.AddWithValue("@ad", paramListe[2]);
290             cmdUpdate.Parameters.AddWithValue("@soyad", paramListe[3]);
291             cmdUpdate.Parameters.AddWithValue("@mail", paramListe[4]);
292             cmdUpdate.Parameters.AddWithValue("@telNo", paramListe[5]);
293             cmdUpdate.Parameters.AddWithValue("@durumu", paramListe[6]);
294             cmdUpdate.Parameters.AddWithValue("@Id", txtIdInfo.Text);
295
296             cmdUpdate.ExecuteNonQuery(); // execute işlemi
297
298             MessageBox.Show("Müşteri Başarıyla Güncellendi.");
299             getAllCustomersData(); // datagridview refresh
300
301         } catch (Exception ex)
302         {
303
304             MessageBox.Show(ex.Message);
305         }
306         finally
307         {
308             baglanti.Close(); //bağlantıyı kapat
309         }
310     }
```

Textboxların alabileceği değerlerin kontrolleri için gereken kod blokları,

```
312     // textboxların alabileceği değerlerin kontrolü
313     1 reference
314     private void txtTcInfo_KeyPress(object sender, KeyPressEventArgs e)
315     {
316         e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar); // sadece sayısal giriş
317     }
318     1 reference
319     private void txtEhliyetInfo_KeyPress(object sender, KeyPressEventArgs e)
320     {
321         if (!char.IsLetterOrDigit(e.KeyChar) && e.KeyChar != '\b') // sayısal ve harf girişi - boşluk hariç
322         {
323             e.Handled = true;
324         }
325     }
326     1 reference
327     private void txtMusteriAdiInfo_KeyPress(object sender, KeyPressEventArgs e)
328     {
329         e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
330             && !char.IsSeparator(e.KeyChar); // sadece harf girişi
331     }
332     1 reference
333     private void txtMusteriSoyadiInfo_KeyPress(object sender, KeyPressEventArgs e)
334     {
335         e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
336             && !char.IsSeparator(e.KeyChar); // sadece harf girişi
337     }
338     1 reference
339     private void txtTelefonNoInfo_KeyPress(object sender, KeyPressEventArgs e)
340     {
341         e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar); // sadece sayısal giriş
```

Textboxların validating eventleri içerisindeki kodlar,

```
343 // *** Validate işlemleri ***
344 private void txtTcInfo_Validating(object sender, CancelEventArgs e)
345 {
346     if (string.IsNullOrWhiteSpace(txtTcInfo.Text)) //text null veya boşluk ise
347     {
348         errorProvider.SetError(txtTcInfo, "Lütfen TC kimlik numarasını girin."); // hata mesajı yazdır
349     }
350     else
351     {
352         errorProvider.SetError(txtTcInfo, null); //hata mesajını kaldır
353     }
354 }
355
356 private void txtEhliyetInfo_Validating(object sender, CancelEventArgs e)
357 {
358     if (string.IsNullOrWhiteSpace(txtEhliyetInfo.Text)) //text null veya boşluk ise
359     {
360         errorProvider.SetError(txtEhliyetInfo, "Lütfen ehliyet numarasını girin."); // hata mesajı yazdır
361     }
362     else
363     {
364         errorProvider.SetError(txtEhliyetInfo, null); // hata mesajını kaldır
365     }
366 }
367
368 private void txtMusteriAdiInfo_Validating(object sender, CancelEventArgs e)
369 {
370     if (string.IsNullOrWhiteSpace(txtMusteriAdiInfo.Text)) //text null veya boşluk ise
371     {
372         errorProvider.SetError(txtMusteriAdiInfo, "Lütfen müşteri adını girin."); // hata mesajı yazdır
373     }
374     else
375     {
376         errorProvider.SetError(txtMusteriAdiInfo, null); // hata mesajını kaldır
377     }
378 }
379
380 private void txtMusteriSoyadiInfo_Validating(object sender, CancelEventArgs e)
381 {
382     if (string.IsNullOrWhiteSpace(txtMusteriSoyadiInfo.Text)) //text null veya boşluk ise
383     {
384         errorProvider.SetError(txtMusteriSoyadiInfo, "Lütfen müşteri soyadını girin."); // hata mesajını yazdır
385     }
386     else
387     {
388         errorProvider.SetError(txtMusteriSoyadiInfo, null); // hata mesajını kaldır
389     }
390 }
391
392 private void txtEpostaInfo_Validating(object sender, CancelEventArgs e)
393 {
394     string email = txtEpostaInfo.Text;
395     bool isValid = IsValidEmail(email); // IsValidEmail() methodu ile girilen verinin eposta olup olmaması durumuna göre true-false döner
396     if (isValid) //true ise
397     {
398         errorProvider.SetError(txtEpostaInfo, null); // hata mesajını kaldır
399     }
400     else
401     {
402         errorProvider.SetError(txtEpostaInfo, "Lütfen geçerli bir e-posta adresi girin"); // hata mesajını yazdır
403     }
404 }
405
406 private void txtTelefonNoInfo_Validating(object sender, CancelEventArgs e)
407 {
408     if (string.IsNullOrWhiteSpace(txtTelefonNoInfo.Text)) //text null veya boşluk ise
409     {
410         errorProvider.SetError(txtTelefonNoInfo, "Lütfen telefon numarası girin."); // hata mesajını yazdır
411     }
412     else
413     {
414         errorProvider.SetError(txtTelefonNoInfo, null); // hata mesajını kaldır
415     }
416 }
417
418 private void txtAddTc_Validating(object sender, CancelEventArgs e)
419 {
420     if (string.IsNullOrWhiteSpace(txtAddTc.Text) || txtAddTc.Text == "Tc Kimlik No")
421     {
422         errorProvider.SetError(txtAddTc, "Lütfen TC kimlik numarasını girin."); // hata mesajını yazdır
423     }
424     else
425     {
426         errorProvider.SetError(txtAddTc, null); // hata mesajını kaldır
427     }
428 }
```

```

430    1 reference
431    private void txtAddEhliyetNo_Validating(object sender, CancelEventArgs e)
432    {
433        if (string.IsNullOrWhiteSpace(txtAddEhliyetNo.Text) || txtAddEhliyetNo.Text == "Ehliyet No")
434        {
435            errorProvider.SetError(txtAddEhliyetNo, "Lütfen ehliyet numarasını girin."); // hata mesajını yazdır
436        }
437        else
438        {
439            errorProvider.SetError(txtAddEhliyetNo, null); // hata mesajını kaldır
440        }
441    }
442    1 reference
443    private void txtAddMusteriAdi_Validating(object sender, CancelEventArgs e)
444    {
445        if (string.IsNullOrWhiteSpace(txtAddMusteriAdi.Text) || txtAddMusteriAdi.Text == "Müşteri Adı")
446        {
447            errorProvider.SetError(txtAddMusteriAdi, "Lütfen müşteri adını girin."); // hata mesajını yazdır
448        }
449        else
450        {
451            errorProvider.SetError(txtAddMusteriAdi, null); // hata mesajını kaldır
452        }
453    }
454    1 reference
455    private void txtAddMusteriSoyadi_Validating(object sender, CancelEventArgs e)
456    {
457        if (string.IsNullOrWhiteSpace(txtAddMusteriSoyadi.Text) || txtAddMusteriSoyadi.Text == "Müşteri Soyadı")
458        {
459            errorProvider.SetError(txtAddMusteriSoyadi, "Lütfen müşteri soyadını girin."); // hata mesajını yazdır
460        }
461        else
462        {
463            errorProvider.SetError(txtAddMusteriSoyadi, null); // hata mesajını kaldır
464        }
465    }
466    1 reference
467    private void txtAddEposta_Validating(object sender, CancelEventArgs e)
468    {
469        string email = txtAddEposta.Text;
470        bool isValid = IsValidEmail(email);
471        if (isValid)
472        {
473            errorProvider.SetError(txtAddEposta, null); // hata mesajını kaldır
474        }
475        else if (email == "E-posta")
476        {
477            errorProvider.SetError(txtAddEposta, "Lütfen bir e-posta adresi girin"); // hata mesajı ver
478        }
479        else
480        {
481            errorProvider.SetError(txtAddEposta, "Lütfen geçerli bir e-posta adresi girin"); // hata mesajı ver
482        }
483    }
484
485    1 reference
486    private void txtAddNumara_Validating(object sender, CancelEventArgs e)
487    {
488        if (string.IsNullOrWhiteSpace(txtAddNumara.Text) || txtAddNumara.Text == "Telefon No")
489        {
490            errorProvider.SetError(txtAddNumara, "Lütfen telefon numarası girin."); // hata mesajı ver
491        }
492        else
493        {
494            errorProvider.SetError(txtAddNumara, null); // hata mesajını kaldır
495        }
496    }
497    1 reference
498    private void cmbBoxDurumu_Validating(object sender, CancelEventArgs e)
499    {
500        if (cmbBoxDurumu.SelectedItem.ToString() == "Seçiniz...")
501        {
502            errorProvider.SetError(cmbBoxDurumu, "Lütfen bir seçim yapın."); // hata mesajı ver
503        }
504        else
505        {
506            errorProvider.SetError(cmbBoxDurumu, null); // hata mesajını kaldır
507        }
508    }

```

isEmpty() methodu içerisindeki kodlar,

```
508     1 reference
509     private bool isEmpty() // ilgili textboxlar boş mu kontrolünü yapmak için boş ise true döner
510     {
511         bool IsEmpty = false;
512         if (
513             txtAddTc.Text == "Tc Kimlik No" ||
514             txtAddEhliyetNo.Text == "Ehliyet No" ||
515             txtAddMusteriAdi.Text == "Müşteri Adı" ||
516             txtAddMusteriSoyadi.Text == "Müşteri Soyadı" ||
517             txtAddEposta.Text == "E-posta" ||
518             txtAddNumara.Text == "Telefon No" ||
519             cmbBoxDurumu.SelectedItem.ToString() == "Seçiniz...")
520         {
521             IsEmpty = true;
522         }
523
524     }
525
526     return IsEmpty;
```

Menü içerisindeki butonların click eventleri,

```
528     // Menü ekran geçişleri
529     1 reference
530     private void LnkLabelAracYonetim_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
531     {
532         VehicleManagementScreen vehicleManagementScreen = new VehicleManagementScreen();
533         vehicleManagementScreen.Show();
534         this.Hide();
535     }
536
537     1 reference
538     private void LnkLabelAracTeslim_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
539     {
540         VehicleDeliveryScreen vehicleDeliveryScreen = new VehicleDeliveryScreen();
541         vehicleDeliveryScreen.Show();
542         this.Hide();
543     }
544
545     1 reference
546     private void LnkLabelArackiralamaIslemleri_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
547     {
548         MainScreen mainScreen = new MainScreen();
549         mainScreen.Show();
550         this.Hide();
551     }
552
553     1 reference
554     private void LnkLabelHesapAyar_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
555     {
556         AccountSettingsScreen accountSettingsScreen = new AccountSettingsScreen();
557         accountSettingsScreen.Show();
558         this.Hide();
559     }
560
561     1 reference
562     private void LnkLabelCikis_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
563     {
564         DialogResult result = MessageBox.Show("Hesabınızdan çıkmak istediğinizize emin misiniz?", "Uyarı", MessageBoxButtons.YesNo);
565
566         if (result == DialogResult.Yes)
567         {
568
569             if (this.DialogResult != DialogResult.OK) // Eğer form dialog olarak kullanılmıyorsa
570             {
571                 // Login ekranına yönlendir
572                 LoginScreen loginScreen = new LoginScreen();
573                 loginScreen.Show();
574                 this.Hide();
575             }
576         }
577     }
578 }
```

Textbox placeholder kodları,

```
-- 573 // Textbox placeholder kodları
574 1 reference
575 private void txtAddTc_Enter(object sender, EventArgs e)
576 {
577     if (txtAddTc.Text == "Tc Kimlik No")
578     {
579         txtAddTc.Text = "";
580         txtAddTc.ForeColor = Color.Black;
581     }
582
583 1 reference
584 private void txtAddEhliyetNo_Enter(object sender, EventArgs e)
585 {
586     if (txtAddEhliyetNo.Text == "Ehliyet No")
587     {
588         txtAddEhliyetNo.Text = "";
589         txtAddEhliyetNo.ForeColor = Color.Black;
590     }
591
592 1 reference
593 private void txtAddMusteriAdi_Enter(object sender, EventArgs e)
594 {
595     if (txtAddMusteriAdi.Text == "Müşteri Adı")
596     {
597         txtAddMusteriAdi.Text = "";
598         txtAddMusteriAdi.ForeColor = Color.Black;
599     }
599
600 1 reference
601 private void txtAddMusteriSoyadi_Enter(object sender, EventArgs e)
602 {
603     if (txtAddMusteriSoyadi.Text == "Müşteri Soyadı")
604     {
605         txtAddMusteriSoyadi.Text = "";
606         txtAddMusteriSoyadi.ForeColor = Color.Black;
607     }
608
609
610 1 reference
611 private void txtAddEposta_Enter(object sender, EventArgs e)
612 {
613     if (txtAddEposta.Text == "E-posta")
614     {
615         txtAddEposta.Text = "";
616         txtAddEposta.ForeColor = Color.Black;
617     }
618
619 1 reference
620 private void txtAddNumara_Enter(object sender, EventArgs e)
621 {
622     if (txtAddNumara.Text == "Telefon No")
623     {
624         txtAddNumara.Text = "";
625         txtAddNumara.ForeColor = Color.Black;
626     }
626
627 1 reference
628 private void txtAddTc_Leave(object sender, EventArgs e)
629 {
630     if (txtAddTc.Text == "")
631     {
632         txtAddTc.Text = "Tc Kimlik No";
633         txtAddTc.ForeColor = Color.Gray;
634     }
635
636
637 1 reference
638 private void txtAddEhliyetNo_Leave(object sender, EventArgs e)
639 {
640     if (txtAddEhliyetNo.Text == "")
641     {
642         txtAddEhliyetNo.Text = "Ehliyet No";
643         txtAddEhliyetNo.ForeColor = Color.Gray;
644     }
645
646 1 reference
647 private void txtAddMusteriAdi_Leave(object sender, EventArgs e)
648 {
649     if (txtAddMusteriAdi.Text == "")
650     {
651         txtAddMusteriAdi.Text = "Müşteri Adı";
652         txtAddMusteriAdi.ForeColor = Color.Gray;
653     }
653
```

```

655     1 reference
656     private void txtAddMusteriSoyadi_Leave(object sender, EventArgs e)
657     {
658         if (txtAddMusteriSoyadi.Text == "")
659         {
660             txtAddMusteriSoyadi.Text = "Müşteri Soyadı";
661             txtAddMusteriSoyadi.ForeColor = Color.Gray;
662         }
663     }
664     1 reference
665     private void txtAddEposta_Leave(object sender, EventArgs e)
666     {
667         if (txtAddEposta.Text == "")
668         {
669             txtAddEposta.Text = "E-posta";
670             txtAddEposta.ForeColor = Color.Gray;
671         }
672     }
673     1 reference
674     private void txtAddNumara_Leave(object sender, EventArgs e)
675     {
676         if (txtAddNumara.Text == "")
677         {
678             txtAddNumara.Text = "Telefon No";
679             txtAddNumara.ForeColor = Color.Gray;
680         }
681     }

```

“Müşteri Ekle” butonuna tıklanınca çalışacak kod bloğu,

```

684     1 reference
685     private void btnMusteriEkle_Click(object sender, EventArgs e)
686     {
687         //herhangi bir hata var mı
688         if (
689             errorProvider.GetError(txtAddTc) == "" &&
690             errorProvider.GetError(txtAddEhliyetNo) == "" &&
691             errorProvider.GetError(txtAddMusteriAdi) == "" &&
692             errorProvider.GetError(txtAddMusteriSoyadi) == "" &&
693             errorProvider.GetError(txtAddEposta) == "" &&
694             errorProvider.GetError(txtAddNumara) == "" &&
695             errorProvider.GetError(cmbBoxDurumu) == ""
696         )
697         {
698             if (isEmpty()) //textboxlar boş mu
699             {
700                 MessageBox.Show("Lütfen tüm bilgileri eksiksiz girin.");
701             }
702             else
703             {
704                 try
705                 {
706                     if (baglanti.State == ConnectionState.Open)
707                     {
708                         baglanti.Close();
709                     }
710                     baglanti.Open(); //bağlantıyı aç
711                     // check tc no
712                     SqlCommand cmdCheckTc = new SqlCommand("select * from tblMusteri where Tc = @tc", baglanti); // sqlCommand
713                     cmdCheckTc.Parameters.AddWithValue("@tc", txtAddTc.Text); // parametre geçiriliyor
714                     // SqlDataReader reader = cmdCheckTc.ExecuteReader();
715                     int result = Convert.ToInt32(cmdCheckTc.ExecuteScalar()); //execute ve kaç sonuç döndüğünün result değişkenine atanması
716                     if (result > 0) //ilgili tc ye ait kayıt bulunduysa
717                     {
718                         MessageBox.Show("Eşleşen Kayıt bulundu.");
719                     }
720                 }
721                 else
722                 {
723                     CultureInfo cultureInfo = CultureInfo.CurrentCulture.TextInfo; // metnin ilk harfi büyük gerisi küçük yazdırılmak için
724                     string query =
725                         "insert into tblMusteri " +
726                         "([Tc], [EhliyetNo], [Ad], [Soyad], [Mail], [TelNo], [Durumu]) " +
727                         "Values (@tc,@ehliyetNo,@ad,@soyad,@mail,@telNo,@durumu)" // sql sorgu cümlesi
728                     SqlCommand cmd = new SqlCommand(query, baglanti); // sqlCommand oluşturma
729                     // parametre geçiriliyor...
730                     cmd.Parameters.AddWithValue("@tc", txtAddTc.Text);
731                     cmd.Parameters.AddWithValue("@ehliyetNo", txtAddEhliyetNo.Text.ToUpper());
732                     cmd.Parameters.AddWithValue("@ad", textInfo.ToTitleCase(txtAddMusteriAdi.Text));
733                     cmd.Parameters.AddWithValue("@soyad", textInfo.ToTitleCase(txtAddMusteriSoyadi.Text));
734                     cmd.Parameters.AddWithValue("@mail", txtAddEposta.Text);
735                     cmd.Parameters.AddWithValue("@telNo", txtAddNumara.Text);
736                     cmd.Parameters.AddWithValue("@durumu", cmbBoxDurumu.SelectedItem.ToString() == "Aktif" ? 1 : 0);
737                     cmd.ExecuteNonQuery(); // execute
738                     MessageBox.Show("Müşteri başarıyla eklendi."); // başarılı mesajı
739                     getAllCustomersData(); // datagridview refresh
740                     clearTextbox(); // textboxları temizle
741                 }
742             }
743         }
744     }

```

```

745     catch (Exception ex)
746     {
747         MessageBox.Show("Hata oluştu. Hata Mesajı : "+ex.Message);
748     }
749     finally
750     {
751         baglanti.Close(); // bağlantıyi kapat
752     }
753 }
754 }
755 }
756 }
757 }
758 }
759 else { MessageBox.Show("Lütfen hataları giderin."); }
760 }

```

“Müşteriyi Sil” butonuna tıklanınca çalışacak kod bloğu,

```

762 1 reference
763 private void btnMusteriSil_Click(object sender, EventArgs e)
764 {
765     // onay mesajı
766     DialogResult result = MessageBox.Show(txtIdInfo.Text +
767         "\n nolu müsteriyi silmek istediğinizde emin misiniz? Bu işlem geri alınamaz.", "Uyarı", MessageBoxButtons.YesNo);
768
769     if (result == DialogResult.Yes) // onay verildiye
770     {
771         try
772         {
773             if (baglanti.State == ConnectionState.Closed)
774             {
775                 baglanti.Open();
776             }
777
778             // ARACI SİL
779             string query = "delete from tblMusteri where Id = @Id"; // sql sorgu cümlesi
780             SqlCommand cmd = new SqlCommand(query, baglanti); // sqlCommand oluşturma
781             cmd.Parameters.AddWithValue("@Id", txtIdInfo.Text); // parametre geçme
782             cmd.ExecuteNonQuery(); // execute işlemi
783             MessageBox.Show(txtIdInfo.Text + " nolu müsteri silindi."); // başarılı mesajı
784             getAllCustomersData(); // datagridview refresh
785             lblMusteriYonetim.Focus();
786         }
787         catch (Exception ex)
788         {
789             MessageBox.Show(ex.Message);
790         }
791         finally
792         {
793             baglanti.Close(); // bağlantıyi kapat
794         }
795     }
796 }

```

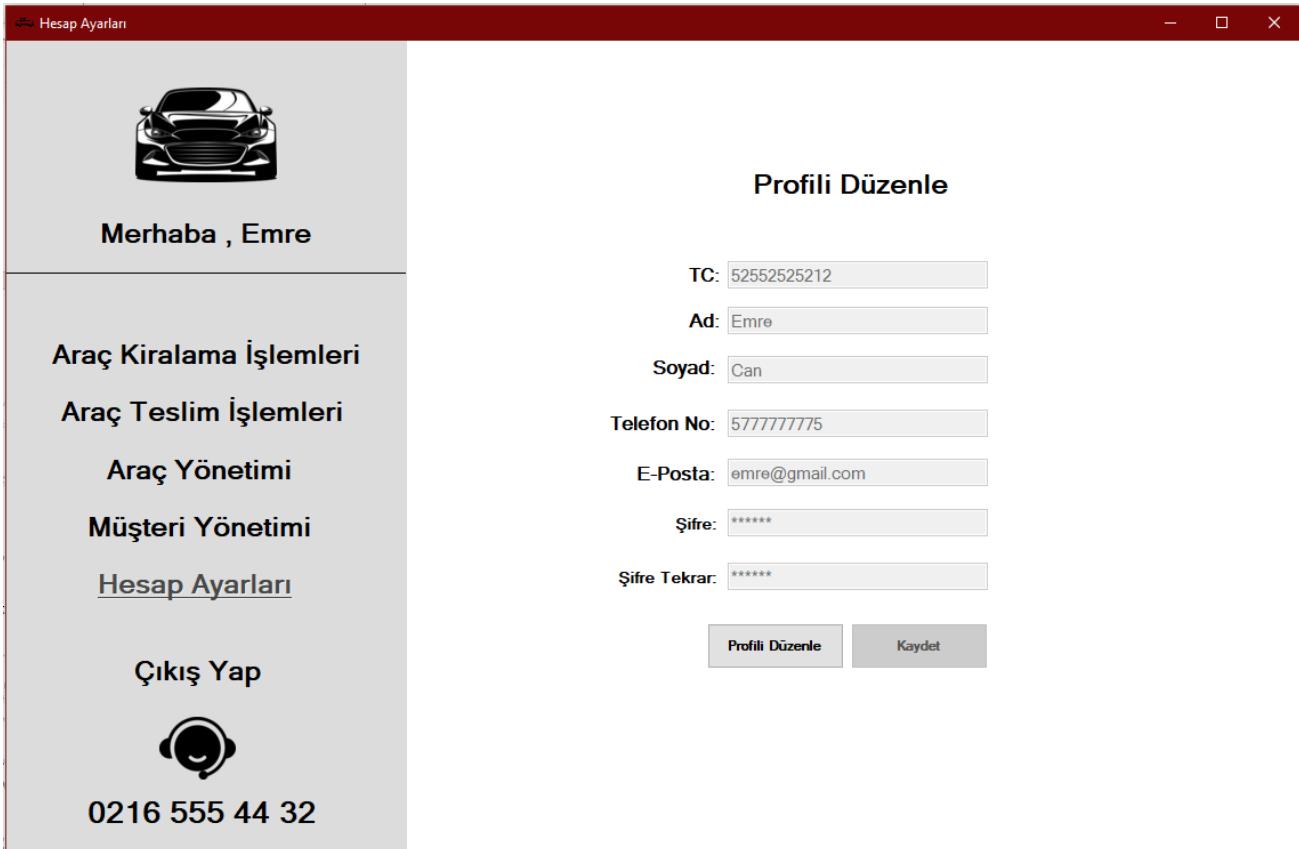
clearTextbox() methodu içerisindeki kodlar,

```

798 1 reference
799 private void clearTextbox() // ilgili textboxlar içerisindeki veriyi temizler ve placeholder larını yazar
800 {
801     txtAddTc.Text = "Tc Kimlik No";
802     txtAddTc.ForeColor = Color.Gray;
803     txtAddEhliyetNo.Text = "Ehliyet No";
804     txtAddEhliyetNo.ForeColor = Color.Gray;
805     txtAddMusteriAdi.Text = "Müşteri Adı";
806     txtAddMusteriAdi.ForeColor = Color.Gray;
807     txtAddMusteriSoyadi.Text = "Müşteri Soyadı";
808     txtAddMusteriSoyadi.ForeColor = Color.Gray;
809     txtAddEposta.Text = "E-posta";
810     txtAddEposta.ForeColor = Color.Gray;
811     txtAddNumara.Text = "Telefon No";
812     txtAddNumara.ForeColor = Color.Gray;
813     cmbBoxDurumu.SelectedIndex= 0;

```

E8-Hesap Ayarları Ekranı



Kodları

Veritabanı bağlantısı , ErrorProvider nesnesinin bağlanması ve form çağrıldığında çalışacak methodlar:

```
19 // veritabanı connection işlemi
20 SqlConnection bağlantı = new SqlConnection("Data Source=DESKTOP-BTIS91A;Initial Catalog=Db_Arackiralama;Integrated Security=True");
21
22 // error provider nesnesi oluşturuluyor
23 ErrorProvider(errorProvider = new ErrorProvider());
24
25 List<String> oldData = new List<String>();
26 List<String> newData = new List<String>();
27 4 references
28 public AccountSettingsScreen()
29 {
30     InitializeComponent();
31
32     // Karşılama Mesajı
33     lblLeftName.Text = "Merhaba , " + LoginScreen.userName;
34
35     getMemberInfo();
36 }
```

setEnableTextbox() ve setDisableTextbox() methodları içerisindeki kodlar,

```
38     1 reference
39     public void setEnableTextbox() // ilgili textboxları veri girişine açar
40     {
41         txtTcKimlik.Enabled = true;
42         txtAd.Enabled = true;
43         txtSoyad.Enabled = true;
44         txtNumara.Enabled = true;
45         txtEposta.Enabled = true;
46         txtPassword.Enabled = true;
47         txtPasswordRepeat.Enabled = true;
48     }
49     2 references
50     public void setDisableTextbox() // ilgili textboxları veri girişine kapatır
51     {
52         txtTcKimlik.Enabled = false;
53         txtAd.Enabled = false;
54         txtSoyad.Enabled = false;
55         txtNumara.Enabled = false;
56         txtEposta.Enabled = false;
57         txtPassword.Enabled = false;
58         txtPasswordRepeat.Enabled = false;
59     }
```

FormClosing eventi içerisindeki kodlar,

```
61     1 reference
62     private void AccountSettingsScreen_FormClosing(object sender, FormClosingEventArgs e)
63     {
64         if (e.CloseReason == CloseReason.UserClosing) // Form, kullanıcı tarafından kapatıldıysa
65         {
66             DialogResult result = MessageBox.Show("Uygulamadan çıkmak istediğinizden emin misiniz?", "Uyarı", MessageBoxButtons.YesNo);
67
68             if (result == DialogResult.Yes)
69             {
70                 e.Cancel = false;
71
72                 if (this.DialogResult != DialogResult.OK) // Eğer form dialog olarak kullanılmıyorsa
73                 {
74                     // Login ekranına yönlendir
75                     LoginScreen loginScreen = new LoginScreen();
76                     loginScreen.Show();
77                 }
78             }
79             else
80             {
81                 e.Cancel = true; // Formun kapatılmasını engelle
82             }
83         }
84         else // Diğer kapatılma sebeplerinde uygulamayı kapat
85         {
86             Application.Exit();
87         }
88     }
```

getMemberInfo() methodu içerisindeki kodlar,

```
89     1 reference
90     private void getMemberInfo() // giriş yapılan hesaba ait bilgileri ilgili textlere yazdırır
91     {
92         txtTcKimlik.Text = LoginScreen.tc;
93         txtAd.Text = LoginScreen.userName;
94         txtSoyad.Text = LoginScreen.lastName;
95         txtEposta.Text = LoginScreen.eposta;
96         txtNumara.Text = LoginScreen.telNo;
97         txtPassword.Text = LoginScreen.password;
98         txtPasswordRepeat.Text = LoginScreen.password;
```

fetchOldData() methodu içerisindeki kodlar,

```
121     1 reference
122     public void fetchOldData() //eski verileri ilgili textboxlara yazdırır
123     {
124         txtTcKimlik.Text = oldData[0];
125         txtAd.Text = oldData[1];
126         txtSoyad.Text = oldData[2];
127         txtNumara.Text = oldData[3];
128         txtEposta.Text = oldData[4];
129         txtPassword.Text = oldData[5];
130         txtPasswordRepeat.Text = oldData[6];
131     }
```

“Düzenle” butonuna tıklandığında çalışacak kodlar,

```
1 reference
100 private void btnDuzenle_Click(object sender, EventArgs e) // düzenle butonuna tıklandığında çalışacak kod bloğu
101 {
102     btnDuzenle.Visible = false; // düzenle butonunu gizle
103     btnIptal.Visible = true; // iptal butonunu göster
104     setEnableTextbox(); // textboxları aktif et
105     btnKaydet.Enabled = true; // kaydet butonunu tıklanabilir yap
106     txtPassword.PasswordChar = '\u0000'; // şifreyi görünür yap
107     txtPasswordRepeat.PasswordChar = '\u0000'; // şifre tekrarını görünür yap
108
109     oldData.Clear(); // listeyi temizle
110     //eski veriyi listeye aktarmak (iptal durumunda bilgileri tekrar getirmek için)
111     oldData.Add(txtTcKimlik.Text);
112     oldData.Add(txtAd.Text);
113     oldData.Add(txtSoyad.Text);
114     oldData.Add(txtNumara.Text);
115     oldData.Add(txtEposta.Text);
116     oldData.Add(txtPassword.Text);
117     oldData.Add(txtPasswordRepeat.Text);
118
119 }
120 }
```

isValidPhoneNumber() ve isValidEmail() methodları içerisindeki kodlar,

```
1 reference
131 private bool IsValidPhoneNumber(string phoneNumber) // Verilen parametre telefon numarası mı kontrol ediliyor
132 {
133     bool isValid = true;
134
135     // Telefon numarasının uzunluğunu kontrol ediliyor.
136     if (phoneNumber.Length != 10)
137     {
138         isValid = false;
139     }
140
141     // Telefon numarasının yalnızca sayılarından oluşanluğu kontrol ediliyor.
142     foreach (char c in phoneNumber)
143     {
144         if (!char.IsDigit(c))
145         {
146             isValid = false;
147             break;
148         }
149     }
150
151     return isValid;
152 }
1 reference
154 private bool IsValidEmail(string email) // Verilen parametre email formatına uygunluğu kontrol ediliyor
155 {
156     if (string.IsNullOrWhiteSpace(email)) // Eğer email boş ya da null ise, true döner
157     {
158         return false;
159     }
160
161     try // Eğer email geçerli bir e-posta formatında değilse, false döndürülür. Aksi takdirde true döndürülür.
162     {
163         MailAddress mailAddress = new MailAddress(email);
164         return true;
165     }
166     catch (FormatException)
167     {
168         return false;
169     }
170 }
```

Hataları temizlemek için kullandığım clearError() methodu içerisindeki kodlar,

```
1 reference
169 public void clearError() // hata mesajlarını siler(iptal butonu için)
170 {
171     errorProvider.SetError(txtTcKimlik, "");
172     errorProvider.SetError(txtAd, "");
173     errorProvider.SetError(txtSoyad, "");
174     errorProvider.SetError(txtNumara, "");
175     errorProvider.SetError(txtEposta, "");
176     errorProvider.SetError(txtPassword, "");
177     errorProvider.SetError(txtPasswordRepeat, "");
178 }
```

Textboxların alabileceği değerlerin kontrolü için gereken kodlar,

```
191 //Textboxların alabileceği değerlerin kontrolü
192 1 reference
193 private void txtTcKimlik_KeyPress(object sender, KeyPressEventArgs e)
194 {
195     e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar); // Yalnızca sayısal değerlerin girilmesine izin verir.
196 }
197
198 1 reference
199 private void txtAd_KeyPress(object sender, KeyPressEventArgs e)
200 {
201     e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
202         && !char.IsSeparator(e.KeyChar); // sadece harf girişi
203 }
204
205 1 reference
206 private void txtSoyad_KeyPress(object sender, KeyPressEventArgs e)
207 {
208     e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
209         && !char.IsSeparator(e.KeyChar); // sadece harf girişi
210 }
211
212 }
```

Textboxların validating eventleri içerisindeki kodlar,

```
215 // *** Validate ***
216 1 reference
217 private void txtTcKimlik_Validating(object sender, CancelEventArgs e)
218 {
219     if (txtTcKimlik.Text.Length < 11)
220     {
221         errorProvider.SetError(txtTcKimlik, "Lütfen geçerli bir tc kimlik no girin"); // hata mesajını göster
222     }
223     else
224     {
225         errorProvider.SetError(txtTcKimlik, ""); // hata mesajını kaldır
226     }
227 }
228
229 1 reference
230 private void txtAd_Validating(object sender, CancelEventArgs e)
231 {
232     if (txtAd.Text.Length < 3)
233     {
234         errorProvider.SetError(txtAd, "Lütfen adınızı eksiksiz girin"); // hata mesajını göster
235     }
236     else
237     {
238         errorProvider.SetError(txtAd, ""); // hata mesajını kaldır
239     }
240
241 1 reference
242 private void txtSoyad_Validating(object sender, CancelEventArgs e)
243 {
244     if (txtSoyad.Text.Length < 3)
245     {
246         errorProvider.SetError(txtSoyad, "Lütfen soyadınızı eksiksiz girin"); // hata mesajını göster
247     }
248     else
249     {
250         errorProvider.SetError(txtSoyad, ""); // hata mesajını kaldır
251     }
252
253 1 reference
254 private void txtNumara_Validating(object sender, CancelEventArgs e)
255 {
256     if (!IsValidPhoneNumber(txtNumara.Text))
257     {
258         errorProvider.SetError(txtNumara, ""); // hata mesajını kaldır
259     }
260     else
261     {
262         // hata mesajı göster
263         errorProvider.SetError(txtNumara, "Lütfen telefon numaranızı doğru girdiğinizden emin olun.Örn: 5554443322");
264     }
265 }
```

```

265     1 reference
266     private void txtEposta_Validate(object sender, CancelEventArgs e)
267     {
268         if (IsValidEmail(txtEposta.Text))
269         {
270             errorProvider.SetError(txtEposta, ""); // hata mesajını kaldır
271         }
272         else
273         {
274             errorProvider.SetError(txtEposta, "Lütfen geçerli bir eposta girin."); // hata mesajını göster
275         }
276     }
277     1 reference
278     private void txtPassword_Validate(object sender, CancelEventArgs e)
279     {
280         string sifre = txtPassword.Text.Trim(); // şifrede boşluk varsa siler
281
282         // Şifre en az 8 karakter uzunluğunda ve en az bir büyük harf, bir küçük harf ve bir rakam içermelidir.
283         string pattern = @"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[a-zA-Z\d]{8,}$";
284
285         if (!Regex.IsMatch(sifre, pattern)) // şifre kurallara uygun değil ise
286         {
287             // hata mesajı göster
288             errorProvider.SetError(txtPassword,
289                         "$Şifre en az 8 karakter uzunluğunda olmalı ve en az bir büyük harf, bir küçük harf ve bir rakam içermelidir.");
290         }
291         else // şifre kurallara uygunsa
292         {
293             errorProvider.SetError(txtPassword, ""); // hata mesajını kaldır
294         }
295     }
296     1 reference
297     private void txtPasswordRepeat_Validate(object sender, CancelEventArgs e)
298     {
299         string sifreTekrar = txtPasswordRepeat.Text.Trim(); // şifre tekrarında boşluk varsa bunu kaldırıp sifreTekrar değişkenine atar
300
301         if (sifreTekrar != txtPassword.Text.Trim()) // şifre tekrarı , şifre ile aynı değilse
302         {
303             errorProvider.SetError(txtPasswordRepeat, "Şifre tekrarı, şifre ile aynı olmalıdır."); // hata mesajı görüntüle
304         }
305         else
306         {
307             errorProvider.SetError(txtPasswordRepeat, ""); // hata mesajını kaldır
308         }
309     }

```

fetchNewData() methodu içerisindeki kodlar,

```

310     1 reference
311     public void fetchNewData() // yeni verileri ilgili textboxlara yazdırır
312     {
313         txtTcKimlik.Text = newData[0];
314         txtAd.Text = newData[1];
315         txtSoyad.Text = newData[2];
316         txtNumara.Text = newData[3];
317         txtEposta.Text = newData[4];
318         txtPassword.Text = newData[5];
319         txtPasswordRepeat.Text = newData[5];

```

“Kaydet” butonuna tıklanınca çalışacak kod bloğu,

```
1 reference
321     private void btnKaydet_Click(object sender, EventArgs e) // kaydet butonuna tıklanınca yapılacak işlemler
322     {
323         // hata yoksa
324         if (
325             errorProvider.GetError(txtTcKimlik) == ""
326             && errorProvider.GetError(txtAd) == ""
327             && errorProvider.GetError(txtSoyad) == ""
328             && errorProvider.GetError(txtNumara) == ""
329             && errorProvider.GetError(txtEposta) == ""
330             && errorProvider.GetError(txtPassword) == ""
331             && errorProvider.GetError(txtPasswordRepeat) == "")
332         {
333             try
334             {
335                 //update işlemi
336                 if (baglanti.State == ConnectionState.Closed) // bağlantı kapalıysa
337                 {
338                     baglanti.Open(); // bağlantıyi aç
339                 }
340                 // sql sorgu cümlesi
341                 string query = "update tblUye SET Tc=@tc,Ad=@ad,Soyad=@soyad,TelNo=@telNo,Mail=@mail,Sifre=@sifre where Tc = @conTc";
342                 SqlCommand cmd = new SqlCommand(query, baglanti); // sqlCommand oluşturma
343                 //parametre geçiriliyor...
344                 cmd.Parameters.AddWithValue("@tc", txtTcKimlik.Text);
345                 cmd.Parameters.AddWithValue("@ad", txtAd.Text);
346                 cmd.Parameters.AddWithValue("@soyad", txtSoyad.Text);
347                 cmd.Parameters.AddWithValue("@telNo", txtNumara.Text);
348                 cmd.Parameters.AddWithValue("@mail", txtEposta.Text);
349                 cmd.Parameters.AddWithValue("@sifre", txtPassword.Text);
350                 cmd.Parameters.AddWithValue("@conTc", oldData[0]);
351                 cmd.ExecuteNonQuery(); // update execute
352                 MessageBox.Show("Güncelleme başarılı."); // başarılı mesajı
353
354                 string selectQuery = "select Tc,Ad,Soyad,TelNo,Mail,Sifre from tblUye where Tc = @conTc"; //sorgu cümlesi oluşturma
355                 SqlCommand cmd2 = new SqlCommand(selectQuery, baglanti); // sqlCommand oluşturma
356                 cmd2.Parameters.AddWithValue("@conTc", txtTcKimlik.Text); // parametre geçme
357                 SqlDataReader reader = cmd2.ExecuteReader(); // execute işlemi ve dönen verinin reader a atılması
358                 while (reader.Read()) // reader içerisindeki verinin okunması
359                 {
360                     // yeni verilerin newData listesine eklenmesi
361                     newData.Add(reader.GetString(0));
362                     newData.Add(reader.GetString(1));
363                     newData.Add(reader.GetString(2));
364                     newData.Add(reader.GetString(3));
365                     newData.Add(reader.GetString(4));
366                     newData.Add(reader.GetString(5));
367                 }
368
369
370                 reader.Close(); // reader kapat
371                 fetchNewData(); // yeni bilgileri ilgili textboxlara yazdır
372
373                 setDisableTextbox(); // textboxları disable yapar
374                 btnDuzenle.Visible = true; // düzenle butonunu göster
375                 btnIptal.Visible = false; // iptal butonunu gizle
376                 btnKaydet.Enabled= false; // kaydet butonunun tıklanabilir özelliğini kaldır
377             }
378         }
379         catch(Exception ex)
380         {
381             MessageBox.Show(ex.Message);
382         }
383         finally
384         {
385             baglanti.Close(); // bağlantıyi kapat
386         }
387     }
```

Menüdeki LinkLabelların click eventleri içerisindeki kodlar,

```
396     // Menü ekran geçişleri
397     1 reference
398     private void LnkLabelAracKiralamaIslemleri_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
399     {
400         this.Hide();
401         MainScreen mainScreen = new MainScreen();
402         mainScreen.Show();
403     }
404     1 reference
405     private void LnkLabelAracTeslim_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
406     {
407         this.Hide();
408         VehicleDeliveryScreen vehicleDeliveryScreen = new VehicleDeliveryScreen();
409         vehicleDeliveryScreen.Show();
410     }
```

```
411 1 reference
412  private void LnkLabelAracYonetim_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
413  {
414      this.Hide();
415      VehicleManagementScreen vehicleManagementScreen = new VehicleManagementScreen();
416      vehicleManagementScreen.Show();
417  }
418 1 reference
419  private void LnkLabelMusteriYonetim_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
420  {
421      this.Hide();
422      CustomerManagementScreen customerManagementScreen = new CustomerManagementScreen();
423      customerManagementScreen.Show();
424  }
425 1 reference
426  private void LnkLabelCikis_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
427  {
428      DialogResult result = MessageBox.Show("Hesabınızdan çıkmak istediğinizize emin misiniz?", "Uyarı", MessageBoxButtons.YesNo);
429      if (result == DialogResult.Yes)
430      {
431
432          if (this.DialogResult != DialogResult.OK) // Eğer form dialog olarak kullanılmiyorsa
433          {
434              // Login ekranına yönlendir
435              LoginScreen loginScreen = new LoginScreen();
436              loginScreen.Show();
437              this.Hide();
438          }
439      }
440  }
441
```

E9-Admin Ekranı

The screenshot shows the Admin Panel application interface. On the left, there's a sidebar with a car icon, the greeting "Merhaba , admin", a "Üye Yönetimi" link, a "Çıkış Yap" link with a user icon, and the phone number "0216 555 44 32". The main area is titled "Üye Yönetimi" and contains two forms: "Üye Bilgileri" and "Üye Ekle". The "Üye Bilgileri" form shows details for user ID 1: ID: 1, TC: 52552525212, Ad: Emre, Soyad: Can, E-posta: emre@gmail.com, Şifre: 123456, Yetki Düzeyi: Yönetici. Below it is a table of users:

ID	TC	Ad	Soyad	Telefon No	E-posta	Şifre	Yetki Düzeyi
1	52552525212	Emre	Can	5777777775	emre@gmail.com	123456	1
3	54556454463	Pasif	Üye	5555555555	abc@gmail.com	123	0
13	52795215152	Emre	Can	5078976675	emrecan1907@hotmail.com	123456	1
26	54577455754	beyza	koyulmuş	5555555555	beyzakoyulmus@yahoo.com	Beyza123	1

The "Üye Ekle" form has fields for TC, Ad, Soyad, Telefon No, E-posta, Şifre, and Yetki Düzeyi. Buttons include "Oneyi Düzenle", "Oneyi Sil", "Oneyi Ekle", and "Oneyi Sil".

Kodları

Veritabanı bağlantısı , ErrorProvider nesnesinin bağlanması ve form çağrıldığında çalışacak methodlar:

```

19   SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-BTIS91A;Initial Catalog=Db_Arakiralama;Integrated Security=True");
20   ErrorProvider errorProvider = new ErrorProvider();
21   public AdminScreen()
22   {
23     InitializeComponent();
24
25     // Karşılama Mesajı
26     lblLeftName.Text = "Merhaba , " + LoginScreen.userName;
27     cmbBoxYetkiDuzeyi.SelectedIndex= 0;
28
29     getAllMembersData();
30   }
31 }
```

getSelectedMemberInfo() methodu içerisindeki kodlar,

```

58  private void getSelectedMemberInfo() // seçili üyenin bilgilerini ilgili textboxlara yazdırır
59  {
60    if (dataUye.SelectedRows.Count > 0)
61    {
62      DataGridViewRow selectedRow = dataUye.SelectedRows[0];
63      if (selectedRow.Cells[0].Value != null)
64      {
65        txtIdInfo.Text = selectedRow.Cells[0].Value.ToString();
66        txtTcInfo.Text = selectedRow.Cells[1].Value.ToString();
67        txtAdInfo.Text = selectedRow.Cells[2].Value.ToString();
68        txtSoyadInfo.Text = selectedRow.Cells[3].Value.ToString();
69        txtTelNoInfo.Text = selectedRow.Cells[4].Value.ToString();
70        txtEpostaInfo.Text = selectedRow.Cells[5].Value.ToString();
71        txtSifreInfo.Text = selectedRow.Cells[6].Value.ToString();
72        // 7 nolu indisteki veri 0 ise "Pasif Üye" değilse "Yönetici" olarak ilgili textboxa yazdırır.
73        txtYetkiInfo.Text = selectedRow.Cells[7].Value.ToString() == "0" ? "Pasif Üye":"Yönetici";
74      }
75    }
76  }
77 }
```

getAllMembersData() methodu içerisindeki kodlar,

```
34 4 references
35 private void getAllMembersData() // tüm üyelerin bilgilerini çeker
36 {
37     if (baglanti.State == ConnectionState.Open)
38     { baglanti.Close(); }
39
40     DataTable dataTable = new DataTable(); // datatable nesnesi oluşturuluyor
41     try
42     {
43         baglanti.Open(); // bağlantıyı aç
44         string query = "select * from tblUye where YetkiDuzeyi != 2 "; // sql sorgu cümlesi
45         SqlCommand cmd = new SqlCommand(query, baglanti); // sqlCommand oluşturma
46         SqlDataReader reader = cmd.ExecuteReader(); // execute işlemi ve dönen verinin reader nesnesine aktarılması
47         dataTable.Load(reader); // reader içindeki verinin dataTable'a yüklenmesi
48         reader.Close(); // reader kapat
49         dataUye.DataSource = dataTable; // datatable daki verileri datagridview'e yükle
50     }
51     catch (Exception ex)
52     {
53         MessageBox.Show(ex.Message);
54     }
55     finally { baglanti.Close(); } // bağlantıyı kapat
56 }
```

setEnableTextbox() ve setDisableTextbox methodları içerisindeki kodlar,

```
78 1 reference
79 private void setEnableTextbox() // ilgili textboxları veri girişine açar
80 {
81     txtTcInfo.Enabled = true;
82     txtAdInfo.Enabled = true;
83     txtSoyadInfo.Enabled = true;
84     txtTelNoInfo.Enabled = true;
85     txtEpostaInfo.Enabled = true;
86     txtSifreInfo.Enabled = true;
87 }
88 2 references
89 private void setDisableTextbox() // ilgili textboxları veri girişine kapatır
90 {
91     txtTcInfo.Enabled = false;
92     txtAdInfo.Enabled = false;
93     txtSoyadInfo.Enabled = false;
94     txtTelNoInfo.Enabled = false;
95     txtEpostaInfo.Enabled = false;
96     txtSifreInfo.Enabled = false;
97 }
```

clearTextbox() methodu içerisindeki kodlar,

```
98 1 reference
99 private void clearTextbox() // ilgili textboxları temizler ve placeholderlarını yazar
100 {
101     txtAddTc.Text = "Tc Kimlik No";
102     txtAddTc.ForeColor = Color.Gray;
103     txtAddName.Text = "Ad";
104     txtAddName.ForeColor = Color.Gray;
105     txtAddSoyad.Text = "Soyad";
106     txtAddSoyad.ForeColor = Color.Gray;
107     txtAddTelNo.Text = "Telefon No";
108     txtAddTelNo.ForeColor = Color.Gray;
109     txtAddEpusta.Text = "E-posta";
110     txtAddEpusta.ForeColor = Color.Gray;
111     txtAddSifre.Text = "$ifre";
112     txtAddSifre.ForeColor = Color.Gray;
113     cmbBoxYetkiDuzeyi.SelectedIndex = 0;
114 }
```

isEmpty() methodu içerisindeki kodlar,

```
164 1 reference
165 private bool isEmpty() // ilgili textboxların boş olup olmadığını kontrol eder
166 {
167     bool IsEmpty = false;
168     if (
169         txtAddTc.Text == "Tc Kimlik No" ||
170         txtAddName.Text == "Ad" ||
171         txtAddSoyad.Text == "Soyad" ||
172         txtTelNoInfo.Text == "Telefon No" ||
173         txtAddEpusta.Text == "E-posta" ||
174         txtAddSifre.Text == "$ifre" ||
175         cmbBoxYetkiDuzeyi.SelectedItem.ToString() == "Seçiniz..." )
176     {
177         IsEmpty = true;
178     }
179 }
180
181 return IsEmpty;
182 }
```

updateData() methodu içerisindeki kodlar,

```
115  private void updateData(List<String> paramListe) // parametre geçen listeyi veritabanında ilgili kayıta aktarır
116  {
117      try
118      {
119          if (baglanti.State == ConnectionState.Closed)// bağlantı kapalıysa
120          {
121              baglanti.Open(); // bağlantıyı aç
122          }
123
124
125          // *** SQL UPDATE ***
126
127          string query =
128              "update tblUye SET Tc=@tc, Ad = @ad , Soyad = @soyad , " +
129              " TelNo = @telNo, Mail = @mail ,Sifre=@sifre, YetkiDuzeyi = @yetkiDuzeyi where Id = @Id "; // where koşulu !!!
130          SqlCommand cmdUpdate = new SqlCommand(query, baglanti); // sqlCommand oluşturma
131
132          //parametre geçiriliyor...
133          cmdUpdate.Parameters.AddWithValue("@tc", paramListe[0]);
134          cmdUpdate.Parameters.AddWithValue("@ad", paramListe[1]);
135          cmdUpdate.Parameters.AddWithValue("@soyad", paramListe[2]);
136          cmdUpdate.Parameters.AddWithValue("@telNo", paramListe[3]);
137          cmdUpdate.Parameters.AddWithValue("@mail", paramListe[4]);
138          cmdUpdate.Parameters.AddWithValue("@sifre", paramListe[5]);
139          cmdUpdate.Parameters.AddWithValue("@yetkiDuzeyi", paramListe[6]);
140          cmdUpdate.Parameters.AddWithValue("@Id", txtIdInfo.Text);
141
142          cmdUpdate.ExecuteNonQuery(); //execute
143
144          MessageBox.Show(txtIdInfo.Text+" Nolu Üye Başarıyla Güncellendi."); // başarılı mesajı
145          getAllMembersData(); // tüm üyelerin bilgilerini getir
146
147
148
149      }
150      catch (Exception ex)
151      {
152
153          MessageBox.Show(ex.Message);
154      }
155      finally
156      {
157          baglanti.Close(); // bağlantıyı kapat
158      }
159
160
161
162 }
```

FormClosing eventi kod bloğu,

```
192  1 reference
193  private void AdminScreen_FormClosing(object sender, FormClosingEventArgs e)
194  {
195      if (e.CloseReason == CloseReason.UserClosing) // Form, Kullanıcı tarafından kapatıldıysa
196      {
197          DialogResult result = MessageBox.Show("Uygulamadan çıkmak istedığınıza emin misiniz?", "Uyarı", MessageBoxButtons.YesNo);
198
199          if (result == DialogResult.Yes)
200          {
201              e.Cancel = false;
202
203              if (this.DialogResult != DialogResult.OK) // Eğer form dialog olarak kullanılmiyorsa
204              {
205                  // Login ekranına yönlendir
206                  LoginScreen loginScreen = new LoginScreen();
207                  loginScreen.Show();
208              }
209          }
210          else
211          {
212              e.Cancel = true; // Formun kapatılmasını engelle
213          }
214      }
215      else // Diğer kapatılma sebeplerinde uygulamayı kapat
216      {
217          Application.Exit();
218      }
219  }
```

“Üyeyi Düzenle” butonuna tıklandığında çalışacak kod bloğu,

```
225 1 reference
226 private void btnUyeliDuzenle_Click(object sender, EventArgs e) // üyeyi düzenle butonuna tıklandığında yapılacak işlemler
227 {
228     btnUyeliDuzenle.Visible = false; //düzenle butonunu gizle
229     btnKaydet.Visible = true; // kaydet butonunu göster
230     btnIptalEt.Visible = true; // iptal et butonunu göster
231     setEnableTextbox(); // textboxları veri girişine aç
232     dataUye.Enabled = false; // datagridviewi dondur
233     cmbBoxYetkiEdit.Visible = true; // comboboxı göster
234     txtYetkiInfo.Visible = false; // textboxı gizle
235
236     //OLD DATA FOR EDIT
237     List<String> oldData = new List<string>();
238     oldData.Add(txtIdInfo.Text);
239     oldData.Add(txtTcInfo.Text);
240     oldData.Add(txtAdInfo.Text);
241     oldData.Add(txtSoyadInfo.Text);
242     oldData.Add(txtTelNoInfo.Text);
243     oldData.Add(txtEmailInfo.Text);
244     oldData.Add(txtSifreInfo.Text);
245     oldData.Add(txtYetkiInfo.Text == "Pasif Üye" ? "0":"1");
246
247     // DURUMU COMBOBOX
248     cmbBoxYetkiEdit.Items.Clear();
249     cmbBoxYetkiEdit.Items.Add("Yönetici");
250     cmbBoxYetkiEdit.Items.Add("Pasif Üye");
251     switch (oldData[7])
252     {
253         case "1":
254             {
255                 cmbBoxYetkiEdit.SelectedIndex = 0;
256                 break;
257             }
258         default:
259             {
260                 cmbBoxYetkiEdit.SelectedIndex = 1;
261                 break;
262             }
263     }
}
```

“İptal Et” butonuna tıklanınca çalışacak kodlar,

```
265 1 reference
266 private void btnIptalEt_Click(object sender, EventArgs e) // iptal et butonuna tıklanınca yapılacak işlemler
267 {
268     btnUyeliDuzenle.Visible = true; // düzenle butonunu göster
269     btnKaydet.Visible = false; // kaydet butonun gizle
270     lblUyeYonetim.Focus();
271     btnIptalEt.Visible = false; // iptal et butonunu gizle
272     setDisableTextbox(); // textboxları veri girişine kapat
273     dataUye.Enabled = true; // datagridviewi aktif et
274     cmbBoxYetkiEdit.Visible = false; // comboboxı gizle
275     txtYetkiInfo.Visible = true; // textboxı göster
276     // hata mesajlarını kaldır
277     errorProvider.SetError(txtTcInfo, "");
278     errorProvider.SetError(txtAdInfo, "");
279     errorProvider.SetError(txtSoyadInfo, "");
280     errorProvider.SetError(txtTelNoInfo, "");
281     errorProvider.SetError(txtEmailInfo, "");
282     getSelectedMemberInfo(); // seçili üyenin bilgilerini getir
283 }
```

Datagridview SelectionChanged eventi içerisindeki kodlar,

```
220 1 reference
221 private void dataUye_SelectionChanged(object sender, EventArgs e) // farklı bir satır seçildiğinde çalışacak event
222 {
223     getSelectedMemberInfo(); // seçili üyenin bilgilerini getir
224 }
```

“Kaydet” butonuna tıklanınca çalışacak kod bloğu,

```
1 reference
285 private void btnKaydet_Click(object sender, EventArgs e) // kaydet butonuna tıklanınca yapılacak işlemler
286 {
287     // hata mesajı yoksa
288     if (
289         errorProvider.GetError(txtTcInfo) == ""
290         && errorProvider.GetError(txtAdInfo) == ""
291         && errorProvider.GetError(txtSoyadInfo) == ""
292         && errorProvider.GetError(txtTelNoInfo) == ""
293         && errorProvider.GetError(txtEpostaInfo) == ""
294         && errorProvider.GetError(txtSifreInfo) == "")
295     {
296
297         //dataMusteri.ReadOnly = true;
298         dataUye.Enabled = true;
299         btnUyeDuzenle.Visible = true;
300         btnKaydet.Visible = false;
301         btnIptalEt.Visible = false;
302         setDisableTextbox();
303         cmbBoxYetkiEdit.Visible = false;
304         txtYetkiInfo.Visible = true;
305
306         // Yeni veriler
307         List<String> newData = new List<string>();
308         newData.Add(txtTcInfo.Text);
309         newData.Add(txtAdInfo.Text);
310         newData.Add(txtSoyadInfo.Text);
311         newData.Add(txtTelNoInfo.Text);
312         newData.Add(txtEpostaInfo.Text);
313         newData.Add(txtSifreInfo.Text);
314         switch (cmbBoxYetkiEdit.SelectedItem.ToString())
315         {
316             case "Yönetici":
317                 {
318                     newData.Add("1");
319                     break;
320                 }
321
322             default:
323                 {
324                     newData.Add("0");
325                     break;
326                 }
327         }
328
329         updateData(newData);    //sql güncelleştirme
330     }
331 else
332 {
333     MessageBox.Show("Lütfen hataları düzeltin.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
334 }
335 }
```

“Üyeyi Sil” butonuna tıklanınca çalışacak kod bloğu,

```
1 reference
337 private void btnUyeSil_Click(object sender, EventArgs e) // Üye sil butonuna tıklanınca yapılacak işlemler
338 {
339     // onay mesajı
340     DialogResult result = MessageBox.Show(txtIdInfo.Text +
341         " nolu üye silmek istediğinizde emin misiniz? Bu işlem geri alınamaz.", "Uyarı", MessageBoxButtons.YesNo);
342
343     if (result == DialogResult.Yes) // onay verildiyse
344     {
345         try
346         {
347             if (baglanti.State == ConnectionState.Closed) // bağlantı kapalıysa
348             {
349                 baglanti.Open(); // bağlantıyi aç
350             }
351
352             // ARACI SİL
353             string query = "delete from tblUye where Id = @Id"; // sql soru cümlesi
354             SqlCommand cmd = new SqlCommand(query, baglanti); // sqlCommand oluşturma
355             cmd.Parameters.AddWithValue("@Id", txtIdInfo.Text); // parametre geçme
356             cmd.ExecuteNonQuery(); // execute
357             MessageBox.Show(txtIdInfo.Text + " nolu üye silindi."); // başarılı mesajı
358             getAllMembersData(); // datagridview refresh
359             lblUyeYonetim.Focus();
360         }
361
362         catch (Exception ex)
363         {
364             MessageBox.Show(ex.Message);
365         }
366         finally
367         {
368             baglanti.Close(); // bağlantıyi kapat
369         }
370     }
371 }
372 }
```

Textboxlarin placeholder kodlari,

```
374 // Textbox placeholder kodlari
375 1 reference
376 private void txtAddTc_Enter(object sender, EventArgs e)
377 {
378     if (txtAddTc.Text == "Tc Kimlik No")
379     {
380         txtAddTc.Text = "";
381         txtAddTc.ForeColor = Color.Black;
382     }
383 }
384 1 reference
385 private void txtAddTc_Leave(object sender, EventArgs e)
386 {
387     if (txtAddTc.Text == "")
388     {
389         txtAddTc.Text = "Tc Kimlik No";
390         txtAddTc.ForeColor = Color.Gray;
391     }
392 }
393 1 reference
394 private void txtAddName_Enter(object sender, EventArgs e)
395 {
396     if (txtAddName.Text == "Ad")
397     {
398         txtAddName.Text = "";
399         txtAddName.ForeColor = Color.Black;
400     }
401 }
402 1 reference
403 private void txtAddSoyad_Enter(object sender, EventArgs e)
404 {
405     if (txtAddSoyad.Text == "Soyad")
406     {
407         txtAddSoyad.Text = "";
408         txtAddSoyad.ForeColor = Color.Black;
409     }
410 }
411 1 reference
412 private void txtAddTelNo_Enter(object sender, EventArgs e)
413 {
414     if (txtAddTelNo.Text == "Telefon No")
415     {
416         txtAddTelNo.Text = "";
417         txtAddTelNo.ForeColor = Color.Black;
418     }
419 }
420 1 reference
421 private void txtAddEposta_Enter(object sender, EventArgs e)
422 {
423     if (txtAddEposta.Text == "E-posta")
424     {
425         txtAddEposta.Text = "";
426         txtAddEposta.ForeColor = Color.Black;
427     }
428 }
429 1 reference
430 private void txtAddSifre_Enter(object sender, EventArgs e)
431 {
432     if (txtAddSifre.Text == "Şifre")
433     {
434         txtAddSifre.Text = "";
435         txtAddSifre.ForeColor = Color.Black;
436     }
437 }
438 1 reference
439 private void txtAddName_Leave(object sender, EventArgs e)
440 {
441     if (txtAddName.Text == "")
442     {
443         txtAddName.Text = "Ad";
444         txtAddName.ForeColor = Color.Gray;
445     }
446 }
447 1 reference
448 private void txtAddSoyad_Leave(object sender, EventArgs e)
449 {
450     if (txtAddSoyad.Text == "")
451     {
452         txtAddSoyad.Text = "Soyad";
453         txtAddSoyad.ForeColor = Color.Gray;
454     }
455 }
```

```

456 1 reference
457  private void txtAddTelNo_Leave(object sender, EventArgs e)
458  {
459      if (txtAddTelNo.Text == "")
460      {
461          txtAddTelNo.Text = "Telefon No";
462          txtAddTelNo.ForeColor = Color.Gray;
463      }
464  }
465 1 reference
466  private void txtAddEposta_Leave(object sender, EventArgs e)
467  {
468      if (txtAddEposta.Text == "")
469      {
470          txtAddEposta.Text = "E-posta";
471          txtAddEposta.ForeColor = Color.Gray;
472      }
473  }
474 1 reference
475  private void txtAddSifre_Leave(object sender, EventArgs e)
476  {
477      if (txtAddSifre.Text == "")
478      {
479          txtAddSifre.Text = "Şifre";
480          txtAddSifre.ForeColor = Color.Gray;
481      }
482  }

```

Textboxların validating eventleri içerisindeki kodlar,

```

483 // *** Validate ***
484 1 reference
485  private void txtAddTc_Validating(object sender, CancelEventArgs e)
486  {
487      if (string.IsNullOrWhiteSpace(txtAddTc.Text) || txtAddTc.Text == "Tc Kimlik No")
488      {
489          errorProvider.SetError(txtAddTc, "Lütfen TC kimlik numarasını girin."); // hata mesajını yazdır
490      }
491      else
492      {
493          errorProvider.SetError(txtAddTc, null); // hata mesajını kaldır
494      }
495  }
496 1 reference
497  private void txtAddName_Validating(object sender, CancelEventArgs e)
498  {
499      if (string.IsNullOrWhiteSpace(txtAddName.Text) || txtAddName.Text == "Ad")
500      {
501          errorProvider.SetError(txtAddName, "Lütfen üye adını girin."); // hata mesajı yazdır
502      }
503      else
504      {
505          errorProvider.SetError(txtAddName, null); // hata mesajını kaldır
506      }
507  }
508 1 reference
509  private void txtAddSoyad_Validating(object sender, CancelEventArgs e)
510  {
511      if (string.IsNullOrWhiteSpace(txtAddSoyad.Text) || txtAddSoyad.Text == "Soyad")
512      {
513          errorProvider.SetError(txtAddSoyad, "Lütfen üye soyadını girin."); // hata mesajı yazdır
514      }
515      else
516      {
517          errorProvider.SetError(txtAddSoyad, null); // hata mesajını kaldır
518      }
519  }
520 1 reference
521  private void txtAddTelNo_Validating(object sender, CancelEventArgs e)
522  {
523      if (string.IsNullOrWhiteSpace(txtAddTelNo.Text) || txtAddTelNo.Text == "Telefon No")
524      {
525          errorProvider.SetError(txtAddTelNo, "Lütfen telefon numarası girin."); // hata mesajı yazdır
526      }
527      else
528      {
529          errorProvider.SetError(txtAddTelNo, null); // hata mesajını kaldır
530      }
531  }

```

```

546 1 reference
547 private void txtAddEposta_Validate(object sender, CancelEventArgs e)
548 {
549     string email = txtAddEposta.Text;
550     bool isValid = IsValidEmail(email);
551     if (isValid)
552     {
553         errorProvider.SetError(txtAddEposta, null); // hata mesajını kaldır
554     }
555     else if (email == "E-posta")
556     {
557         errorProvider.SetError(txtAddEposta, "Lütfen bir e-posta adresi girin"); // hata mesajı yazdır
558     }
559     else
560     {
561         errorProvider.SetError(txtAddEposta, "Lütfen geçerli bir e-posta adresi girin"); // hata mesajı yazdır
562     }
563 }
564 1 reference
565 private void txtAddSifre_Validate(object sender, CancelEventArgs e)
566 {
567     string sifre = txtAddSifre.Text.Trim(); // şifrede boşluk varsa siler
568     // Şifre en az 8 karakter uzunluğunda ve en az bir büyük harf, bir küçük harf ve bir rakam içermelidir.
569     string pattern = @"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d){8,}$";
570
571     if (!Regex.IsMatch(sifre, pattern)) // şifre kurallara uygun değil ise
572     {
573         // hata mesajı göster
574         errorProvider.SetError(txtAddSifre,
575             "$şifre en az 8 karakter uzunluğunda olmalı ve en az bir büyük harf, bir küçük harf ve bir rakam içermelidir.");
576     }
577     else // şifre kurallara uygunsa
578     {
579
580         errorProvider.SetError(txtAddSifre, ""); // hata mesajını kaldır
581     }
582 }
583 1 reference
584 private void cmbBoxYetkiDuzeyi_Validate(object sender, CancelEventArgs e)
585 {
586     if(cmbBoxYetkiDuzeyi.SelectedIndex==0)
587     {
588         errorProvider.SetError(cmbBoxYetkiDuzeyi,"Lütfen bir seçim yapın"); // hata mesajını yazdır
589     }
590     else
591     {
592         errorProvider.SetError(cmbBoxYetkiDuzeyi, "");// hata mesajını kaldır
593     }
594 }
595 1 reference
596 private void txtTcInfo_Validate(object sender, CancelEventArgs e)
597 {
598     if (txtTcInfo.Text == "" || txtTcInfo.Text.Length < 11)
599     {
600         errorProvider.SetError(txtTcInfo, "Lütfen TC kimlik No Girin"); // hata mesajını yazdır
601     }
602     else
603     {
604         errorProvider.SetError(txtTcInfo, "");// hata mesajını kaldır
605     }
606 }
607 1 reference
608 private void txtAdInfo_Validate(object sender, CancelEventArgs e)
609 {
610     if (txtAdInfo.Text == "" || txtAdInfo.Text.Length < 3)
611     {
612         errorProvider.SetError(txtAdInfo, "Lütfen adı girin."); // hata mesajını yazdır
613     }
614     else
615     {
616         errorProvider.SetError(txtAdInfo, ""); // hata mesajını kaldır
617     }
618 }
619 1 reference
620 private void txtSoyadInfo_Validate(object sender, CancelEventArgs e)
621 {
622     if (txtSoyadInfo.Text == "" || txtSoyadInfo.Text.Length < 2)
623     {
624         errorProvider.SetError(txtSoyadInfo, "Lütfen soyadı girin."); // hata mesajını yazdır
625     }
626     else
627     {
628         errorProvider.SetError(txtSoyadInfo, ""); // hata mesajını kaldır
629     }
}

```

```

631    1 reference
632    private void txtTelNoInfo_Validating(object sender, CancelEventArgs e)
633    {
634        if (txtTelNoInfo.Text == "" || txtTelNoInfo.Text.Length < 10)
635        {
636            errorProvider.SetError(txtTelNoInfo, "Lütfen numara girin."); // hata mesajını yazdır
637        }
638        else
639        {
640            errorProvider.SetError(txtTelNoInfo, ""); // hata mesajını kaldır
641        }
642    }
643    1 reference
644    private void txtEpostaInfo_Validating(object sender, CancelEventArgs e)
645    {
646        string email = txtEpostaInfo.Text;
647        bool isValid = IsValidEmail(email);
648        if (isValid)
649        {
650            errorProvider.SetError(txtEpostaInfo, null); // hata mesajını kaldır
651        }
652        else
653        {
654            errorProvider.SetError(txtEpostaInfo, "Lütfen geçerli bir e-posta adresi girin"); // hata mesajını yazdır
655        }
656    }
657    1 reference
658    private void txtSifreInfo_Validating(object sender, CancelEventArgs e)
659    {
660        string sifre = txtSifreInfo.Text.Trim(); // şifrede boşluk varsa siler
661
662        // Şifre en az 8 karakter uzunluğunda ve en az bir büyük harf, bir küçük harf ve bir rakam içermelidir.
663        string pattern = @"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)[a-zA-Z\d]{8,}$";
664
665        if (!Regex.IsMatch(sifre, pattern)) // şifre kurallara uygun değil ise
666        {
667            // hata mesajı göster
668            errorProvider.SetError(txtSifreInfo,
669                "$şifre en az 8 karakter uzunluğunda olmalı ve en az bir büyük harf, bir küçük harf ve bir rakam içermelidir.");
670        }
671        else // şifre kurallara uygunsu
672        {
673            errorProvider.SetError(txtSifreInfo, ""); // hata mesajını kaldır
674        }
675    }

```

“Üye Ekle” butonuna tıklanınca çalışacak kod blogu,

```

677    1 reference
678    private void btnUyeyiEkle_Click(object sender, EventArgs e) // Üyeyi ekle butonuna tıklandığında yapılacak işlemler
679    {
680        // hata yoksa
681        if (
682            errorProvider.GetError(txtAddTc) == "" &&
683            errorProvider.GetError(txtAddName) == "" &&
684            errorProvider.GetError(txtAddSoyad) == "" &&
685            errorProvider.GetError(txtAddTelNo) == "" &&
686            errorProvider.GetError(txtAddEposta) == "" &&
687            errorProvider.GetError(txtAddSifre) == "" &&
688            errorProvider.GetError(cmbBoxYetkiDuzeyi) == ""
689        )
690        {
691            if (isEmpty()) // boş textbox var ise
692            {
693                MessageBox.Show("Lütfen tüm bilgileri eksiksiz girin.");
694            }
695            else // yok ise
696            {
697                try
698                {
699                    if (baglanti.State == ConnectionState.Open)
700                    {
701                        baglanti.Close();
702                    }
703                    baglanti.Open(); // bağlantıyi aç
704                    // check tc no
705                    SqlCommand cmdCheckTc = new SqlCommand("select * from tblUye where Tc = @tc", baglanti);
706                    cmdCheckTc.Parameters.AddWithValue("@tc", txtAddTc.Text); // parametre geçme
707                    // SqlDataReader reader = cmdCheckTc.ExecuteReader();
708                    int result = Convert.ToInt32(cmdCheckTc.ExecuteScalar()); // kaç adet sonuç döndürüyüne result değişkenine ata
709                    if (result > 0) // sonuç varsa
710                    {
711                        MessageBox.Show("Eşleşen Kayıt bulundu.");
712                    }
713                }
714            }
715        }
716    }

```

```

713     else // sonuç yoksa
    {
714
715         TextInfo textInfo = CultureInfo.CurrentCulture.TextInfo; // metnin ilk harfi büyük gerisi küçük yazdırma için
716         string query =
717             "insert into tblUye ([Tc], [Ad], [Soyad], [TelNo],[Mail], [Sifre],[YetkiDuzeyi]) " +
718             "Values (@tc,@ad,@soyad,@telNo,@mail,@sifre,@yetkiDuzeyi)"; // sql sorgu cümlesi
719         SqlCommand cmd = new SqlCommand(query, baglanti); // sqlCommand oluşturma
720         // parametre geçiriliyor...
721         cmd.Parameters.AddWithValue("@tc", txtAddTc.Text);
722         cmd.Parameters.AddWithValue("@ad", textInfo.ToTitleCase(txtAddName.Text));
723         cmd.Parameters.AddWithValue("@soyad", textInfo.ToTitleCase(txtAddSoyad.Text));
724         cmd.Parameters.AddWithValue("@telNo", txtAddTelNo.Text);
725         cmd.Parameters.AddWithValue("@mail", txtAddEposta.Text);
726         cmd.Parameters.AddWithValue("@sifre", txtAddSifre.Text);
727         cmd.Parameters.AddWithValue("@yetkiDuzeyi", cmbBoxYetkiDuzeyi.SelectedItem.ToString() == "Yönetici" ? 1 : 0);
728         cmd.ExecuteNonQuery(); // execute
729         MessageBox.Show("Üyə başarıyla eklendi."); // başarılı mesajı
730         getAllMembersData(); // datagridview refresh
731         clearTextbox(); // textboxları temizle
732     }
733
734 }
735
736 catch (Exception ex)
737 {
738     MessageBox.Show("Hata oluştu. Hata Mesajı : " + ex.Message);
739 }
740 finally
741 {
742
743     baglanti.Close(); // bağlantıyi kapat
744 }
745
746 }
747
748 }
749
750 else { MessageBox.Show("Lütfen hataları giderin."); }
751

```

Textboxların alabileceği değerlerin kontolü için gereken kod blokları,

```

-- // Textboxların alabileceği değerlerin kontrolleri
1 reference
753 private void txtTcInfo_KeyPress(object sender, KeyPressEventArgs e)
754 {
755     e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar); // sadece sayısal giriş
756 }
757
1 reference
758
759 private void txtAdInfo_KeyPress(object sender, KeyPressEventArgs e)
760 {
761     e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
762     && !char.IsSeparator(e.KeyChar); // sadece harf giriş
763 }
764
1 reference
765 private void txtSoyadInfo_KeyPress(object sender, KeyPressEventArgs e)
766 {
767     e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
768     && !char.IsSeparator(e.KeyChar); // sadece harf giriş
769 }
770
1 reference
771 private void txtTelNoInfo_KeyPress(object sender, KeyPressEventArgs e)
772 {
773     e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar); // sadece sayısal giriş
774 }
775
1 reference
776 private void txtAddTelNo_KeyPress(object sender, KeyPressEventArgs e)
777 {
778     e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar); // sadece sayısal giriş
779 }
780
1 reference
781 private void txtAddTc_KeyPress(object sender, KeyPressEventArgs e)
782 {
783     e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar); // sadece sayısal giriş
784 }
785
1 reference
786 private void txtAddName_KeyPress(object sender, KeyPressEventArgs e)
787 {
788     e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
789     && !char.IsSeparator(e.KeyChar); // sadece harf giriş
790 }
791
1 reference
792 private void txtAddSoyad_KeyPress(object sender, KeyPressEventArgs e)
793 {
794     e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
795     && !char.IsSeparator(e.KeyChar); // sadece harf giriş
796 }

```