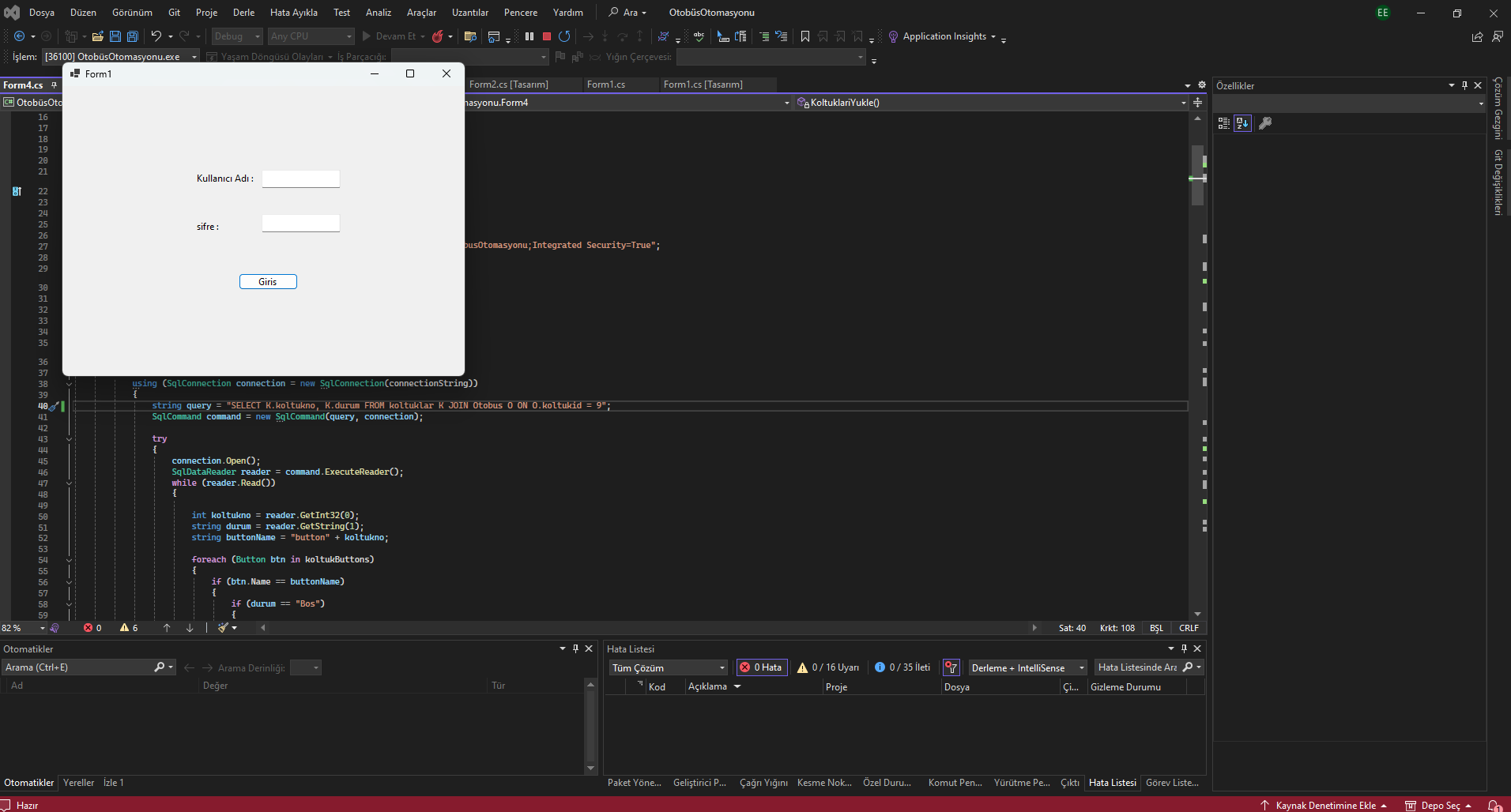
**1.FORM**

****

using System.Data.SqlClient;

namespace OtobüsOtomasyonu

{

public partial class Form1 : Form

{

// Veritabanına bağlanmak için kullanılan bağlantı dizesi

private string connectionString = @"Data Source=DARK;Initial Catalog=OtobusOtomasyonu;Integrated Security=True";

// SqlConnection nesnesi, bağlantıyı yönetmek için kullanılır

private SqlConnection connection;

public Form1()

{

InitializeComponent();

// SqlConnection nesnesini bağlantı dizesi ile başlat

connection = new SqlConnection(connectionString);

}

// Buton tıklama olayı için olay işleyici

private void button1\_Click(object sender, EventArgs e)

{

// Kullanıcı adı ve şifreyi textBox'lardan al

string kadi, ksifre;

kadi = textBox1.Text;

ksifre = textBox2.Text;

// Veritabanı bağlantısını aç

connection.Open();

// Admin kimlik bilgilerini kontrol etmek için SQL komutu oluştur

SqlCommand sorgu = new SqlCommand("SELECT \* FROM Admin where kullanici='" + kadi + "' and sifre='" + ksifre + "'", connection);

// Komutu çalıştır ve sonuçları oku

SqlDataReader oku = sorgu.ExecuteReader();

// Eşleşen bir kayıt bulunursa

if (oku.Read())

{

// Admin formunu göster ve mevcut formu gizle

Form2 admin = new Form2();

admin.Show();

this.Hide();

}

else

{

// Kimlik bilgileri yanlışsa hata mesajı göster

MessageBox.Show("Kullanıcı Adı veya şifre hatalı");

}

// Veritabanı bağlantısını kapat

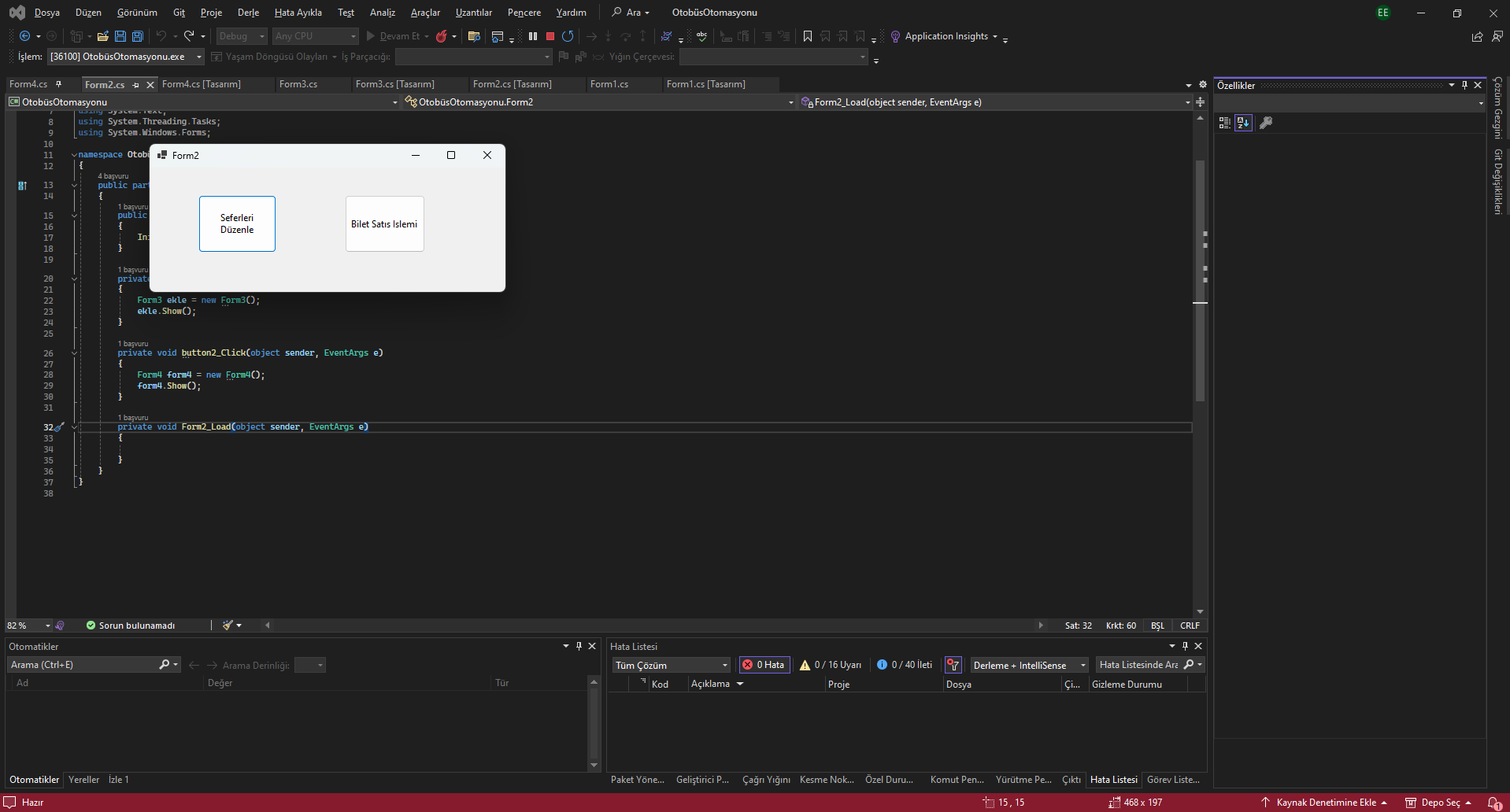
connection.Close();

}

}

}

**2.Form**

****

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace OtobüsOtomasyonu

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

// Ekleme butonuna tıklandığında

private void button1\_Click(object sender, EventArgs e)

{

// Form3'ü oluştur ve göster

Form3 ekle = new Form3();

ekle.Show();

}

// Form4 açma butonuna tıklandığında

private void button2\_Click(object sender, EventArgs e)

{

// Form4'ü oluştur ve göster

Form4 form4 = new Form4();

form4.Show();

}

// Form2 yüklendiğinde

private void Form2\_Load(object sender, EventArgs e)

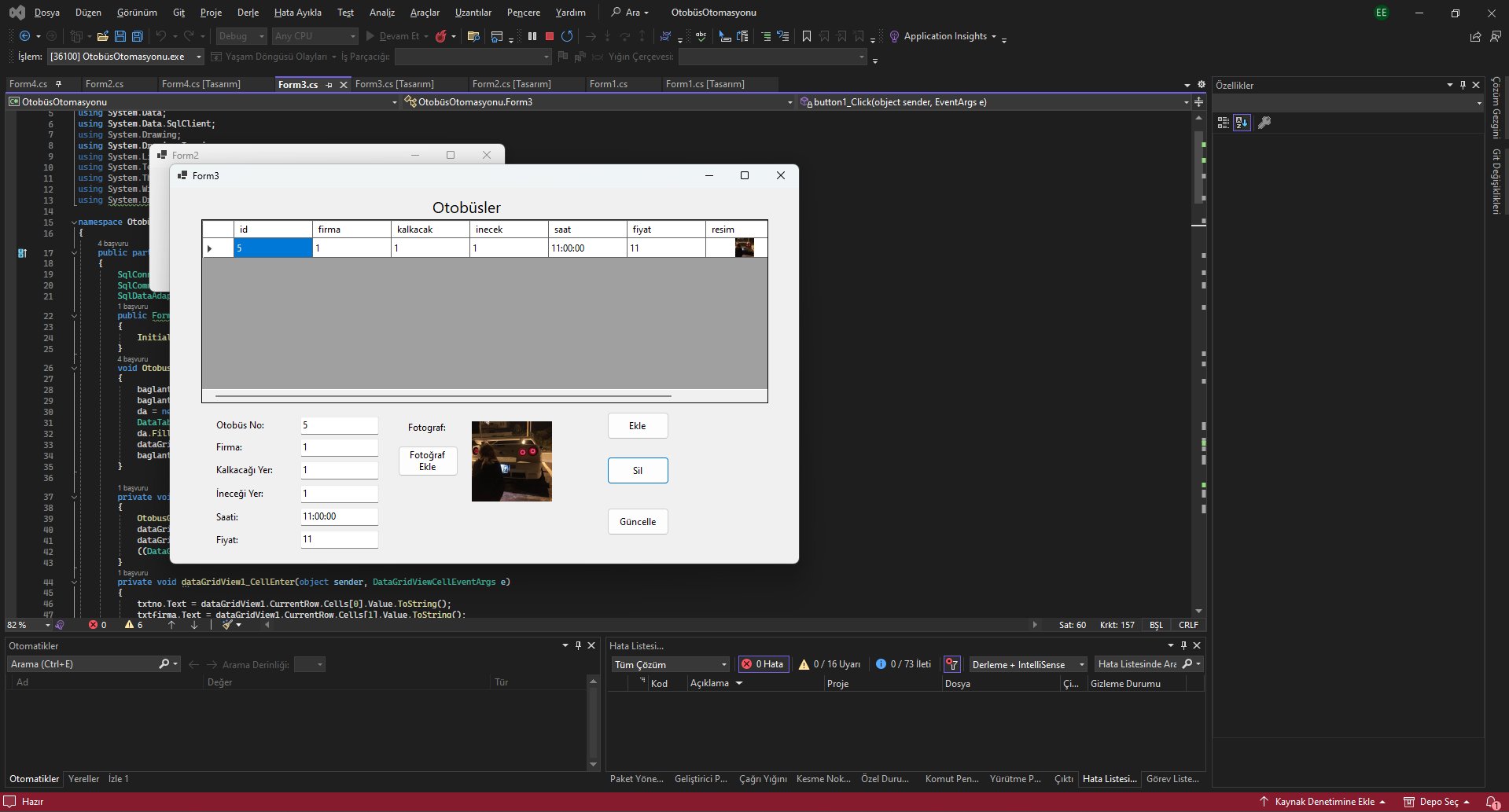
{

}

}

}

**3.Form**

****

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Drawing.Imaging;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace OtobüsOtomasyonu

{

public partial class Form3 : Form

{

SqlConnection baglanti;

SqlCommand komut;

SqlDataAdapter da;

public Form3()

{

InitializeComponent();

}

// Otobüs bilgilerini getir ve DataGridView'e yükle

void OtobusGetir()

{

baglanti = new SqlConnection(@"Data Source=DARK;Initial Catalog=OtobusOtomasyonu;Integrated Security=True");

baglanti.Open();

da = new SqlDataAdapter("SELECT id, firma, kalkacak, inecek, saat, fiyat, resim, koltukid FROM Otobus", baglanti);

DataTable tablo = new DataTable();

da.Fill(tablo);

dataGridView1.DataSource = tablo;

baglanti.Close();

}

// Form3 yüklendiğinde otobüs bilgilerini getir

private void Form3\_Load(object sender, EventArgs e)

{

OtobusGetir();

dataGridView1.Columns["resim"].DefaultCellStyle.NullValue = null;

dataGridView1.Columns["resim"].DefaultCellStyle.Padding = new Padding(-31);

((DataGridViewImageColumn)dataGridView1.Columns["resim"]).ImageLayout = DataGridViewImageCellLayout.Zoom;

}

// DataGridView hücresine girildiğinde metin kutularına değerleri doldur

private void dataGridView1\_CellEnter(object sender, DataGridViewCellEventArgs e)

{

txtno.Text = dataGridView1.CurrentRow.Cells[0].Value.ToString();

txtfirma.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();

txtkalk.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();

txtin.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();

txtsaat.Text = dataGridView1.CurrentRow.Cells[4].Value.ToString();

txtfiyat.Text = dataGridView1.CurrentRow.Cells[5].Value.ToString();

byte[] imageData = (byte[])dataGridView1.CurrentRow.Cells[6].Value;

pictureBox1.Image = ByteArrayToImage(imageData);

}

// Otobüs bilgilerini ekle

private void button1\_Click(object sender, EventArgs e)

{

byte[] imageData = ImageToByteArray(pictureBox1.Image);

string sorgu = "INSERT INTO Otobus(id, firma, kalkacak, inecek, saat, fiyat, resim) VALUES(@id, @firma, @kalkacak, @inecek, @saat, @fiyat, @resim)";

komut = new SqlCommand(sorgu, baglanti);

komut.Parameters.AddWithValue("@id", txtno.Text);

komut.Parameters.AddWithValue("@firma", txtfirma.Text);

komut.Parameters.AddWithValue("@kalkacak", txtkalk.Text);

komut.Parameters.AddWithValue("@inecek", txtin.Text);

komut.Parameters.AddWithValue("@saat", txtsaat.Text);

komut.Parameters.AddWithValue("@fiyat", txtfiyat.Text);

komut.Parameters.Add("@resim", SqlDbType.VarBinary, -1).Value = imageData;

baglanti.Open();

komut.ExecuteNonQuery();

baglanti.Close();

OtobusGetir();

}

// Otobüs bilgilerini sil

private void button2\_Click(object sender, EventArgs e)

{

string sorgu = "DELETE FROM Otobus WHERE id=@id";

komut = new SqlCommand(sorgu, baglanti);

komut.Parameters.AddWithValue("@id", Convert.ToInt32(txtno.Text));

baglanti.Open();

komut.ExecuteNonQuery();

baglanti.Close();

OtobusGetir();

}

// Otobüs bilgilerini güncelle

private void button3\_Click(object sender, EventArgs e)

{

string id = txtno.Text;

string yeniFirma = txtfirma.Text;

string yenikalk = txtkalk.Text;

string yeniinecek = txtin.Text;

string yenisaat = txtsaat.Text;

string yenifiyat = txtfiyat.Text;

byte[] currentImageData = GetImageFromDatabase(Convert.ToInt32(id));

Image eskiresim = ByteArrayToImage(currentImageData);

Image yeniresim = pictureBox1.Image;

byte[] yeniresimBytes;

if (!AreImagesEqual(yeniresim, eskiresim))

{

yeniresimBytes = ImageToByteArray(yeniresim);

}

else

{

yeniresimBytes = currentImageData;

}

string connectionString = "Data Source=DARK;Initial Catalog=OtobusOtomasyonu;Integrated Security=True";

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string kontrolQuery = "SELECT COUNT(\*) FROM Otobus WHERE id = @id";

using (SqlCommand kontrolCommand = new SqlCommand(kontrolQuery, connection))

{

kontrolCommand.Parameters.AddWithValue("@id", id);

int OtobusSayisi = (int)kontrolCommand.ExecuteScalar();

if (OtobusSayisi > 0)

{

string guncellemeQuery = "UPDATE Otobus SET firma = @yeniFirma, kalkacak = @yenikalk, inecek = @yeniinecek, saat = @yenisaat, fiyat = @yenifiyat, resim = @Yeniresim WHERE id = @id";

using (SqlCommand guncellemeCommand = new SqlCommand(guncellemeQuery, connection))

{

guncellemeCommand.Parameters.AddWithValue("@id", id);

guncellemeCommand.Parameters.AddWithValue("@yeniFirma", yeniFirma);

guncellemeCommand.Parameters.AddWithValue("@yenikalk", yenikalk);

guncellemeCommand.Parameters.AddWithValue("@yeniinecek", yeniinecek);

guncellemeCommand.Parameters.AddWithValue("@yenisaat", yenisaat);

guncellemeCommand.Parameters.AddWithValue("@yenifiyat", yenifiyat);

guncellemeCommand.Parameters.Add("@Yeniresim", SqlDbType.VarBinary, -1).Value = yeniresimBytes;

guncellemeCommand.ExecuteNonQuery();

MessageBox.Show("Otobüs bilgileri ve resmi başarıyla güncellendi.");

}

}

else

{

MessageBox.Show("Otobüs bulunamadı. Lütfen geçerli bir otobüs numarası girin.");

}

}

}

OtobusGetir();

}

// Resim seç butonuna tıklandığında

private void btnfoto\_Click(object sender, EventArgs e)

{

OpenFileDialog dosya = new OpenFileDialog();

dosya.Filter = "Resim Dosyası |\*.jpg;\*.nef;\*.png | Tüm Dosyalar |\*.\*";

dosya.ShowDialog();

string dosyayolu = dosya.FileName;

btnfoto.Text = dosyayolu;

pictureBox1.ImageLocation = dosyayolu;

}

// Veritabanından resim al

private byte[] GetImageFromDatabase(int id)

{

byte[] imageData = null;

string connectionString = "Data Source=DARK;Initial Catalog=OtobusOtomasyonu;Integrated Security=True";

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string query = "SELECT resim FROM Otobus WHERE id = @id";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@id", id);

using (SqlDataReader reader = command.ExecuteReader())

{

if (reader.Read())

{

if (!reader.IsDBNull(0))

{

imageData = (byte[])reader[0];

}

}

}

}

}

return imageData;

}

// Resmi byte dizisine dönüştür

private byte[] ImageToByteArray(Image image)

{

using (MemoryStream ms = new MemoryStream())

{

image.Save(ms, ImageFormat.Png);

return ms.ToArray();

}

}

// Byte dizisini resme dönüştür

private Image ByteArrayToImage(byte[] byteArray)

{

using (MemoryStream ms = new MemoryStream(byteArray))

{

Image image = Image.FromStream(ms);

return new Bitmap(image);

}

}

// İki resmin eşit olup olmadığını kontrol et

private bool AreImagesEqual(Image image1, Image image2)

{

if (image1.Width != image2.Width || image1.Height != image2.Height)

{

return false;

}

using (MemoryStream ms1 = new MemoryStream())

using (MemoryStream ms2 = new MemoryStream())

{

image1.Save(ms1, ImageFormat.Png);

image2.Save(ms2, ImageFormat.Png);

byte[] image1Bytes = ms1.ToArray();

byte[] image2Bytes = ms2.ToArray();

return StructuralComparisons.StructuralEqualityComparer.Equals(image1Bytes, image2Bytes);

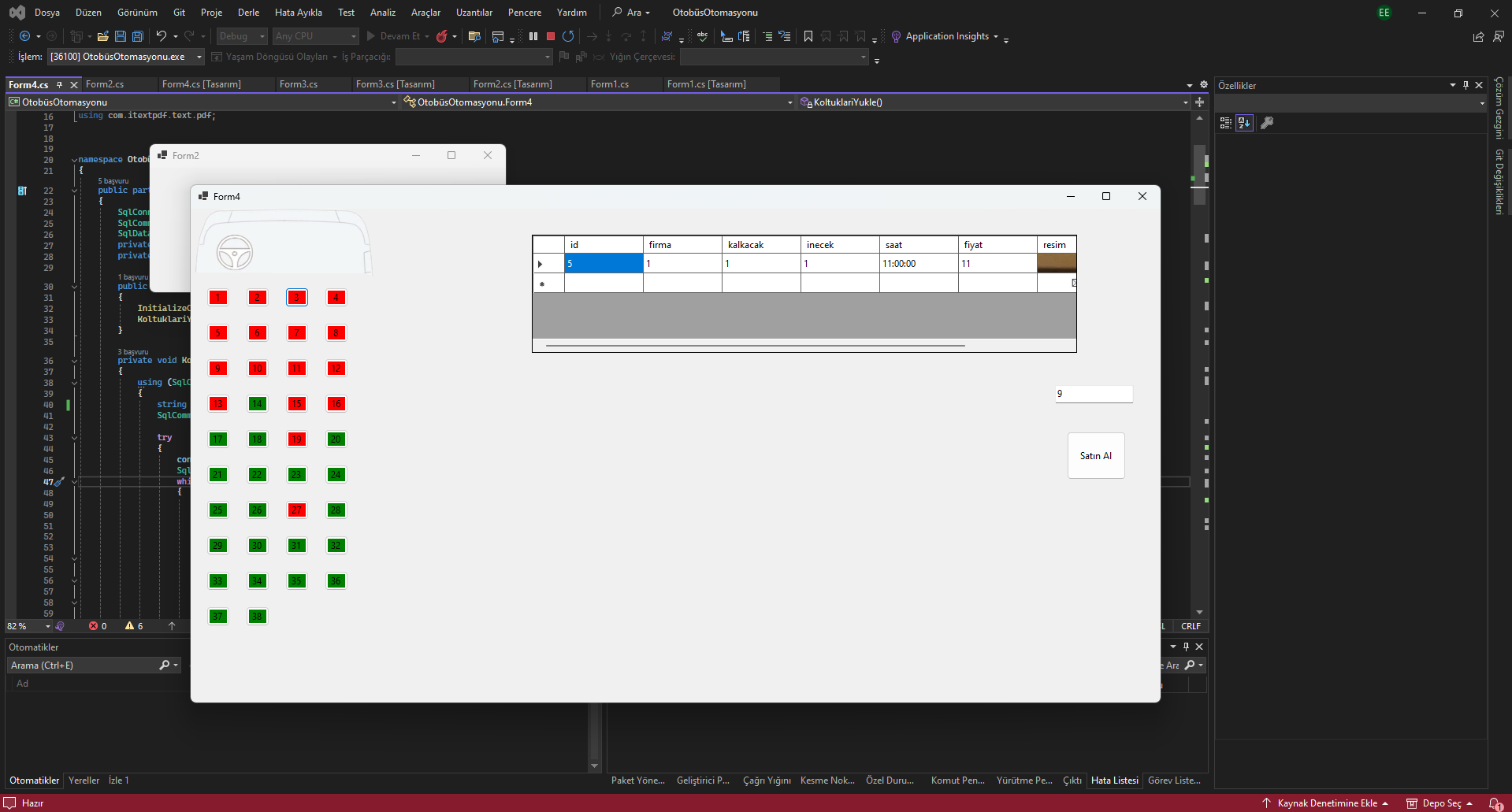
}

}

}

}

**4.Form**

****

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using iTextSharp.text;

using iTextSharp.text.pdf;

using System.IO;

namespace OtobüsOtomasyonu

{

public partial class Form4 : Form

{

SqlConnection baglanti; // SQL bağlantısı

SqlCommand komut; // SQL komutu

SqlDataAdapter da; // SQL Data Adapter

private string connectionString = "Data Source=DARK;Initial Catalog=OtobusOtomasyonu;Integrated Security=True"; // Bağlantı stringi

private List<Button> koltukButtons = new List<Button>(); // Koltuk butonları listesi

public Form4()

{

InitializeComponent();

KoltuklariYukle(); // Koltukları yükle

}

private void KoltuklariYukle()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT K.koltukno, K.durum FROM koltuklar K JOIN Otobus O ON O.koltukid = 9";

SqlCommand command = new SqlCommand(query, connection);

try

{

connection.Open();

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

int koltukno = reader.GetInt32(0); // Koltuk numarasını oku

string durum = reader.GetString(1); // Koltuk durumunu oku

string buttonName = "button" + koltukno;

foreach (Button btn in koltukButtons)

{

if (btn.Name == buttonName)

{

if (durum == "Bos")

{

btn.BackColor = System.Drawing.Color.Green; // Koltuk boş ise yeşil renkte

}

else

{

btn.BackColor = System.Drawing.Color.Red; // Koltuk dolu ise kırmızı renkte

}

break;

}

}

}

}

catch (Exception ex)

{

MessageBox.Show("Bir hata oluştu: " + ex.Message);

}

}

}

private void KoltukButtonClick(object sender, EventArgs e)

{

Button clickedButton = (Button)sender;

int koltukNo = int.Parse(clickedButton.Name.Replace("button", ""));

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "SELECT durum FROM koltuklar WHERE koltukno = @KoltukNo";

SqlCommand command = new SqlCommand(query, connection);

command.Parameters.AddWithValue("@KoltukNo", koltukNo);

connection.Open();

string durum = (string)command.ExecuteScalar();

if (durum == "Dolu")

{

MessageBox.Show($"Koltuk {koltukNo} dolu.");

}

else

{

MessageBox.Show($"Koltuk {koltukNo} boş.");

}

}

}

private void BiletSatinAl(int koltukNo)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string updateQuery = "UPDATE koltuklar SET durum = 'Dolu' WHERE koltukno = @KoltukNo";

SqlCommand updateCommand = new SqlCommand(updateQuery, connection);

updateCommand.Parameters.AddWithValue("@KoltukNo", koltukNo);

connection.Open();

int rowsAffected = updateCommand.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show($"Koltuk {koltukNo} için bilet satın alındı!");

Button btn = koltukButtons.FirstOrDefault(b => b.Name == "button" + koltukNo);

if (btn != null)

{

btn.BackColor = System.Drawing.Color.Red; // Koltuk dolu olduğunda kırmızı renkte

}

// PDF fişi oluştur

PdfFisOlustur(koltukNo);

}

else

{

MessageBox.Show($"Koltuk {koltukNo} için bilet satın alma işlemi başarısız oldu!");

}

}

}

private void PdfFisOlustur(int koltukNo)

{

StringBuilder receipt = new StringBuilder();

receipt.AppendLine("------ Fiş Çıktısı ------");

receipt.AppendLine($"Koltuk No: {koltukNo}");

receipt.AppendLine($"Tarih: {DateTime.Now}");

receipt.AppendLine("-------------------------");

try

{

SaveFileDialog saveFileDialog = new SaveFileDialog();

saveFileDialog.Filter = "PDF Dosyaları|\*.pdf";

saveFileDialog.Title = "PDF Olarak Kaydet";

saveFileDialog.ShowDialog();

if (saveFileDialog.FileName != "")

{

// PDF dokümanı oluştur

Document pdfDoc = new Document();

PdfWriter.GetInstance(pdfDoc, new FileStream(saveFileDialog.FileName, FileMode.Create));

pdfDoc.Open();

// Fiş çıktısını PDF'e yazdır

pdfDoc.Add(new Paragraph(receipt.ToString()));

pdfDoc.Close();

MessageBox.Show("Fiş çıktısı başarıyla PDF olarak kaydedildi: " + saveFileDialog.FileName);

}

}

catch (Exception ex)

{

MessageBox.Show("PDF oluşturma hatası: " + ex.Message);

}

}

private void btnsat\_Click(object sender, EventArgs e)

{

try

{

Button clickedButton = (Button)sender;

int koltukNo = int.Parse(clickedButton.Name.Replace("button", ""));

using (SqlConnection connection = new SqlConnection(connectionString))

{

string checkQuery = "SELECT durum FROM koltuklar WHERE koltukno = @KoltukNo";

SqlCommand checkCommand = new SqlCommand(checkQuery, connection);

checkCommand.Parameters.AddWithValue("@KoltukNo", koltukNo);

connection.Open();

string durum = (string)checkCommand.ExecuteScalar();

if (durum == "Dolu")

{

MessageBox.Show($"Koltuk {koltukNo} zaten satılmış!");

}

else

{

BiletSatinAl(koltukNo);

}

KoltuklariYukle();

}

}

catch (Exception ex)

{

MessageBox.Show($"Hata oluştu: {ex.Message}");

}

}

private void Form4\_Load(object sender, EventArgs e)

{

KoltuklariYukle();

OtobusGetir();

Panel panel = new Panel();

panel.Dock = DockStyle.Fill;

this.Controls.Add(panel);

int koltukSayisi = 38;

int koltukGenislik = 30;

int koltukYukseklik = 25;

int padding = 20;

int yOffset = 80; // Butonları aşağı kaydırmak için ofset

int sutunSayisi = 4;

for (int i = 0; i < koltukSayisi; i++)

{

Button btnSat = new Button();

btnSat.Name = "button" + (i + 1);

btnSat.Text = (i + 1).ToString();

btnSat.Size = new Size(koltukGenislik, koltukYukseklik);

int row = i / sutunSayisi;

int col = i % sutunSayisi;

btnSat.Location = new Point(padding + col \* (koltukGenislik + padding), yOffset + padding + row \* (koltukYukseklik + padding));

btnSat.Click += btnsat\_Click;

panel.Controls.Add(btnSat);

koltukButtons.Add(btnSat);

}

}

private void dataGridView1\_CellEnter(object sender, DataGridViewCellEventArgs e)

{

txtid.Text = dataGridView1.CurrentRow.Cells[7].Value.ToString();

}

void OtobusGetir()

{

baglanti = new SqlConnection(@"Data Source=DARK;Initial Catalog=OtobusOtomasyonu;Integrated Security=True");

baglanti.Open();

da = new SqlDataAdapter("SELECT id, firma, kalkacak, inecek, saat, fiyat, resim, koltukid FROM Otobus", baglanti);

DataTable tablo = new DataTable();

da.Fill(tablo);

dataGridView1.DataSource = tablo;

baglanti.Close();

}

private void IDyeGoreKoltuklariGetir(int id)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

string query = "select koltukno,durum from Otobus O join koltuklar K on O.koltukid = K.koltukid WHERE id = @id";

SqlCommand command = new SqlCommand(query, connection);

command.Parameters.AddWithValue("@id", id);

connection.Open();

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

int koltukNo = reader.GetInt32(0);

string durum = reader.GetString(1);

Button btn = koltukButtons.FirstOrDefault(b => b.Name == "button" + koltukNo);

if (btn != null)

{

if (durum == "Dolu")

{

btn.BackColor = Color.Red;

}

else

{

btn.BackColor = Color.Green;

}

}

}

}

}

private void button19\_Click(object sender, EventArgs e)

{

// Bu kısımda eksik olan işlemler varsa, buraya ekleyebilirsiniz.

}

private void dataGridView1\_SelectionChanged(object sender, EventArgs e)

{

if (dataGridView1.SelectedRows.Count > 0)

{

int selectedID = Convert.ToInt32(dataGridView1.SelectedRows[0].Cells["id"].Value);

IDyeGoreKoltuklariGetir(selectedID);

}

}

}

}