KODLAR

KİLİT EKRANI KODALRI:

```csharp
using Bunifu.UI.WinForms;
using Bunifu.UI.WinForms.Renderers.Snackbar;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace GeleceğeNOTProje
{   //CEMİLCEM OKTAY 222523015
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
            Control[] controls = panel2.Controls.Cast<Control>().ToArray();
            bunifuFormDock1.SubscribeControlsToDragEvents(controls);
        }

        private void bunifuImageButton1_Click(object sender, EventArgs e)
        {
            // Kullanıcı adı ve şifreyi alır ve tabloda karşılaştırır eğer uyuşursa formu kapatım diğer formu açar uyuşmaz ise uyarı verir.
            string kullaniciAdi = bunifuTextBox1.Text;
            string sifre = bunifuTextBox2.Text;
            string connectionString = "Data Source=DESKTOP-242E7PH\\SQLEXPRESS;Initial Catalog=tbl_GelecegeNot;Integrated Security=True";
            string query = "SELECT * FROM kullanici WHERE kullaniciAdi = @kullaniciAdi AND sifre = @sifre";

            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                SqlCommand command = new SqlCommand(query, connection);
                command.Parameters.AddWithValue("@kullaniciAdi", kullaniciAdi);
                command.Parameters.AddWithValue("@sifre", sifre);

                try
                {
                    connection.Open();
                    SqlDataReader reader = command.ExecuteReader();

                    if (reader.Read())
                    {
                        Form1 FrmAna = new Form1();
                        FrmAna.kullanici = kullaniciAdi;

                        FrmAna.Show();
                        this.Hide();
                    }
                    else
                    {
                        MessageBox.Show("Kullanıcı adı veya şifre hatalı.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
                    }
                    reader.Close();
                }
```

```csharp
                }
                catch (Exception ex)
                {
                    MessageBox.Show("Bir hata oluştu: " + ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
                finally
                {
                    connection.Close();
                }
            }
        }

        private void bunifuShadowPanel1_ControlAdded(object sender, ControlEventArgs e)
        {

        }

        private void bunifuImageButton2_Click(object sender, EventArgs e)
        {
            //Kayıt olma ekranını açar. Girişi sağlayan araçları gizler ve kayıt araçlarını gösterir. Tekrar tıklandığında ise işlemin tersini yapar.
            bunifuPictureBox1.Visible = !bunifuPictureBox1.Visible;
            bunifuTextBox3.Visible = !bunifuTextBox3.Visible;
            bunifuPictureBox4.Visible = !bunifuPictureBox4.Visible;
            bunifuTextBox4.Visible = !bunifuTextBox4.Visible;
            label5.Visible = !label5.Visible;
            label4.Visible = !label4.Visible;
            label2.Visible = !label2.Visible;
            label3.Visible = !label3.Visible;
            bunifuImageButton1.Visible = !bunifuImageButton1.Visible;
            bunifuImageButton3.Visible = !bunifuImageButton3.Visible;
            bunifuImageButton4.Visible = !bunifuImageButton4.Visible;
        }

        private void bunifuImageButton3_Click(object sender, EventArgs e)
        {
            // Kullanıcı kaydını yapar. Kullanıcı adı ve şifrenin boş olurp olup olmadığına bakar ve uygunsa tabloya yeni kullanıcıyı kaydeder ve giriş ekranına yönlendirir.
            string kullaniciAdi = bunifuTextBox3.Text;
            string sifre = bunifuTextBox4.Text;

            if (string.IsNullOrWhiteSpace(kullaniciAdi) || string.IsNullOrWhiteSpace(sifre))
            {
                MessageBox.Show("Kullanıcı adı veya şifre boş bırakılamaz.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }

            string connectionString = "Data Source=DESKTOP-242E7PH\\SQLEXPRESS;Initial Catalog=tbl_GelecegeNot;Integrated Security=True";
            string insertUserQuery = "INSERT INTO kullanici (kullaniciAdi, sifre) VALUES (@kullaniciAdi, @sifre)";

            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                SqlCommand insertUserCommand = new SqlCommand(insertUserQuery, connection);
                insertUserCommand.Parameters.AddWithValue("@kullaniciAdi", kullaniciAdi);
                insertUserCommand.Parameters.AddWithValue("@sifre", sifre);

                try
                {
                    connection.Open();
                    int rowsAffected = insertUserCommand.ExecuteNonQuery();

                    if (rowsAffected > 0)
                    {
```

```
118                    {
119                        MessageBox.Show("Kullanıcı başarıyla eklendi.", "Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Information);
120                        bunifuImageButton4_Click(sender, e);
121                        bunifuTextBox3.Text = string.Empty;
122                        bunifuTextBox4.Text = string.Empty;
123                    }
124                    else
125                    {
126                        MessageBox.Show("Kullanıcı eklenemedi. Lütfen tekrar deneyin.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
127                    }
128                }
129                catch (Exception ex)
130                {
131                    MessageBox.Show("Bir hata oluştu: " + ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
132                }
133                finally
134                {
135                    connection.Close();
136                }
137            }
138        }
139
140        private void label6_Click(object sender, EventArgs e)[...]
166
167        private void bunifuImageButton4_Click(object sender, EventArgs e)
168        {
169            //Giriş ekranını açar. Girişi sağlayan araçları gösterir ve kayıt araçlarını gizler. Tekrar tıklandığında ise işlemin tersini yapar.
170            bunifuPictureBox1.Visible = !bunifuPictureBox1.Visible;
171            bunifuTextBox3.Visible = !bunifuTextBox3.Visible;
172            bunifuPictureBox4.Visible = !bunifuPictureBox4.Visible;
173            bunifuTextBox4.Visible = !bunifuTextBox4.Visible;
174            label5.Visible = !label5.Visible;
175            label4.Visible = !label4.Visible;
176            label2.Visible = !label2.Visible;
177            label3.Visible = !label3.Visible;
178            bunifuImageButton1.Visible = !bunifuImageButton1.Visible;
179            bunifuImageButton3.Visible = !bunifuImageButton3.Visible;
180            bunifuImageButton4.Visible = !bunifuImageButton4.Visible;
181        }
182
183        private void Form2_Load(object sender, EventArgs e)
184        {
185            // TODO: Bu kod satırı 'tbl_GeleceğeNotDataSet6.kullanici' tablosuna veri yükler. Bunu gerektiği şekilde taşıyabilir, veya kaldırabilirsiniz.
186            this.kullaniciTableAdapter.Fill(this.tbl_GeleceğeNotDataSet6.kullanici);
187
188        }
189
190        private void label7_Click(object sender, EventArgs e)[...]
194    }
195 }
196
```

ANA EKRAN KODLARI:

```
1   using System;
2   using System.Collections.Generic;
3   using System.ComponentModel;
4   using System.Data;
5   using System.Drawing;
6   using System.Linq;
7   using System.Text;
8   using System.Threading.Tasks;
9   using System.Data.SqlClient;
10  using System.Windows.Forms;
11  using Bunifu.UI.WinForms;
12  using Newtonsoft.Json.Linq;
13  using static System.Windows.Forms.VisualStyles.VisualStyleElement.Tab;
14
15  namespace GeleceğeNOTProje
16  {   //CEMİLCEM OKTAY 222523815
17      public partial class Form1 : Form
18      {
19          public Form1()
20          {
21              InitializeComponent();
22              Control[] controls = panel2.Controls.Cast<Control>().ToArray();
23              bunifuFormDock1.SubscribeControlsToDragEvents(controls);
24              timer1 = new Timer();
25              timer1.Interval = 1000; // 1 saniyede bir tetiklenecek şekilde checkedListBox1
26              timer1.Tick += new EventHandler(timer1_Tick);
27              timer1.Start();
28
29          }
30
31          public string kullanici;
32
33          //DESKTOP-242E7PH\SQLEXPRESS
34          //Data Source=DESKTOP-242E7PH\SQLEXPRESS;Initial Catalog=tbl_GelecegeNot;Integrated Security=True;Trust Server Certificate=True
35          private void panel3_Paint(object sender, PaintEventArgs e)[...]
38
39          private void Form1_Load(object sender, EventArgs e)
40          {
41              // TODO: Bu kod satırı 'tbl_GelecegeNotDataSet5.kullanici' tablosuna veri yükler. Bunu gerektiği şekilde taşıyabilir, veya kaldırabilirsiniz.
42              this.kullaniciTableAdapter.Fill(this.tbl_GelecegeNotDataSet5.kullanici);
43              label16.Text = kullanici;
44              // TODO: Bu kod satırı 'tbl_GelecegeNotDataSet4.Gelecek' tablosuna veri yükler. Bunu gerektiği şekilde taşıyabilir, veya kaldırabilirsiniz.
45              this.gelecekTableAdapter2.Fill(this.tbl_GelecegeNotDataSet4.Gelecek);
46              // TODO: Bu kod satırı 'tbl_GelecegeNotDataSet3.Gecmis' tablosuna veri yükler. Bunu gerektiği şekilde taşıyabilir, veya kaldırabilirsiniz.
47              this.gecmisTableAdapter.Fill(this.tbl_GelecegeNotDataSet3.Gecmis);
48              timer1.Start();
49
50              this.kullaniciTableAdapter.Fill(this.tbl_GelecegeNotDataSet5.kullanici);
51              label16.Text = kullanici;
52
53              LoadUserNotes(kullanici);
54
55              dataGridView1.RowHeadersWidth = 20;
56              dataGridView2.RowHeadersWidth = 20;
57
58              int progressValue = bunifuCircleProgress1.Value;
59              label8.Text = "Gün %" + progressValue.ToString() + " oranında doldu";
60              bunifuImageButton7_Click(sender, e);
61
```

```csharp
            DateTime today = DateTime.Today;
            label18.Text = today.ToString("dd/MM/yyyy") + " - " + today.ToString("dddd");

            string connectionString = "Data Source=DESKTOP-242E7PH\\SQLEXPRESS;Initial Catalog=tbl_GelecegeNot;Integrated Security=True";

            using (SqlConnection connection = new SqlConnection(connectionString))
            { }

            LoadUserNotes(kullanici);
        }

        private void LoadUserNotes(string kullaniciAdi)
        {
            // Veritabanına bağlanıp kullanıcıya ait ID yi alıyor ardından aynı ID ye sahip verileri tablolara yüklüyor.
            string connectionString = "Data Source=DESKTOP-242E7PH\\SQLEXPRESS;Initial Catalog=tbl_GelecegeNot;Integrated Security=True";

            int kullaniciID = GetKullaniciID(kullaniciAdi, connectionString);

            if (kullaniciID != -1)
            {
                LoadDataIntoGridView(kullaniciID, connectionString);
            }
            else
            {
                MessageBox.Show("Kullanıcı bulunamadı.");
            }
        }
        private int GetKullaniciID(string kullaniciAdi, string connectionString)
        {
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                connection.Open();

                string query = "SELECT id FROM kullanici WHERE kullaniciAdi = @kullaniciAdi";

                using (SqlCommand command = new SqlCommand(query, connection))
                {
                    command.Parameters.AddWithValue("@kullaniciAdi", kullaniciAdi);

                    object result = command.ExecuteScalar();
                    return result != null ? Convert.ToInt32(result) : -1;
                }
            }
        }
        private void LoadDataIntoGridView(int kullaniciID, string connectionString)
        {
            string queryGelecek = "SELECT * FROM gelecek WHERE kullaniciID = @kullaniciID ORDER BY atarih ASC";
            string queryGecmis = "SELECT * FROM gecmis WHERE kullaniciID = @kullaniciID ORDER BY tarih ASC";

            // İleriki tarihe olan notları tabloya yazdıran kod.
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                connection.Open();

                using (SqlCommand command = new SqlCommand(queryGelecek, connection))
                {
                    command.Parameters.AddWithValue("@kullaniciID", kullaniciID);
```

```csharp
                    command.Parameters.AddWithValue("@kullaniciID", kullaniciID);
                    using (SqlDataAdapter adapter = new SqlDataAdapter(command))
                    {
                        DataTable dataTable = new DataTable();
                        adapter.Fill(dataTable);
                        dataGridView1.DataSource = dataTable;
                    }
                }
                // Geçmiş tarihte olan notları tabloya yazdıran kod.
                using (SqlCommand command = new SqlCommand(queryGecmis, connection))
                {
                    command.Parameters.AddWithValue("@kullaniciID", kullaniciID);
                    using (SqlDataAdapter adapter = new SqlDataAdapter(command))
                    {
                        DataTable dataTable = new DataTable();
                        adapter.Fill(dataTable);
                        dataGridView2.DataSource = dataTable;
                    }
                }
            }
            // TODO: Bu kod satırı 'tbl_GelecegeNotDataSet2.gelecek' tablosuna veri yükler. Bunu gerektiği şekilde taşıyabilir, veya kaldırabilirsiniz.
            this.gelecekTableAdapter1.Fill(this.tbl_GelecegeNotDataSet2.gelecek);
            // TODO: Bu kod satırı 'tbl_GelecegeNotDataSet1.gelecek' tablosuna veri yükler. Bunu gerektiği şekilde taşıyabilir, veya kaldırabilirsiniz.
            this.gelecekTableAdapter.Fill(this.tbl_GelecegeNotDataSet1.gelecek);
        }

        private void Button1_Click(object sender, EventArgs e)
        {
            //Eklenen notun kullanıcı adınından ID yi bulup ona göre veritabanına atılmasını sağlayan kod
            string not = TxtBoxNOT.Text;
            DateTime tarih;

            string kullaniciAdi = label16.Text;
            int kullaniciID = -1;

            string connectionString = "Data Source=DESKTOP-242E7PH\\SQLEXPRESS;Initial Catalog=tbl_GelecegeNot;Integrated Security=True";

            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                // Kullanıcının Id sini bulan kod.
                string query = "SELECT id FROM kullanici WHERE kullaniciAdi = @kullaniciAdi";
                using (SqlCommand command = new SqlCommand(query, connection))
                {
                    command.Parameters.AddWithValue("@kullaniciAdi", kullaniciAdi);
                    connection.Open();
                    object result = command.ExecuteScalar();
                    if (result != null)
                    {
                        kullaniciID = Convert.ToInt32(result);
                    }
                }

                if (kullaniciID != -1 && DateTime.TryParse(TxtBoxTARİH.Text, out tarih))
                {
                    string sql = "INSERT INTO gelecek (atarih, [not], kullaniciID) VALUES (@tarih, @not, @kullaniciID)";

                    using (SqlCommand command = new SqlCommand(sql, connection))
                    {
```

```csharp
                    {
                        command.Parameters.AddWithValue("@tarih", tarih);
                        command.Parameters.AddWithValue("@not", not);
                        command.Parameters.AddWithValue("@kullaniciID", kullaniciID);
                        command.ExecuteNonQuery();
                    }
                    bunifuImageButton7_Click(sender, e);
                }
                else
                {
                    MessageBox.Show("Tarih seçiniz.");
                }
                connection.Close();
            }
            TxtBoxNOT.Text = null;
        }
        private void monthCalendar1_DateSelected(object sender, DateRangeEventArgs e)
        {
            //TAkvimden seçilen tarihin yazdırılmasını sağlayan kod.
            DateTime selectedDate = e.Start;
            TxtBoxTARIH.Text = selectedDate.ToString("dd/MM/yyyy");
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            // Mevcut zamanı alıp günün % kaçının dolduğunu hesaplayıp yazdıran kod.
            DateTime now = DateTime.Now;

            double secondsInADay = 24 * 60 * 60;
            double secondsSoFar = now.TimeOfDay.TotalSeconds;
            double progress = (secondsSoFar / secondsInADay) * 100;

            bunifuCircleProgress1.Value = (int)Math.Round(progress);
            int progressValue = bunifuCircleProgress1.Value;
            label8.Text = $"Gün %{progressValue} oranında doldu";
            label11.Text = now.ToString("HH:mm:ss");
        }
        private void checkedListBox1_ItemCheck(object sender, ItemCheckEventArgs e)
        {
            //Bugüne ait notların hepsi işaretlenince "tebrikler " mesajı gönderen kod.
            bool allChecked = true;
            for (int i = 0; i < checkedListBox1.Items.Count; i++)
            {
                if (i != e.Index)
                {
                    if (!checkedListBox1.GetItemChecked(i))
                    {
                        allChecked = false;
                        break;
                    }
                }
                else if (e.NewValue == CheckState.Unchecked)
                {
                    allChecked = false;
                    break;
                }
            }

            if (allChecked)
            {
                bunifuPictureBox1.Visible = true;
                label9.Visible = true;

                //Tebrikler mesajının 2 saniye görünür yapıp tekrar gizleyen kod.
                System.Timers.Timer timer = new System.Timers.Timer(2000); // 2 saniye
                timer.Elapsed += (s, ev) =>
                {
                    bunifuPictureBox1.Invoke(new Action(() => bunifuPictureBox1.Visible = false));
                    label9.Invoke(new Action(() => label9.Visible = false));
                    timer.Stop();
                    timer.Dispose();
                };
                timer.Start();
            }
        }

        private void bunifuImageButton1_Click(object sender, EventArgs e)
        {
            // İleriki tarih notlarının olduğu tablodan ve veritabanından seçili satırı silen kod.
            if (dataGridView1.SelectedRows.Count > 0)
            {
                var selectedRow = dataGridView1.SelectedRows[0];
                var selectedNot = selectedRow.Cells["not"].Value.ToString();
                var selectedTarih = Convert.ToDateTime(selectedRow.Cells["atarih"].Value);
                string connectionString = "Data Source=DESKTOP-242E7PH\\SQLEXPRESS;Initial Catalog=tbl_GelecegeNot;Integrated Security=True";

                string queryDeleteRow = "DELETE FROM Gelecek WHERE [not] = @not AND [atarih] = @atarih";

                try
                {
                    using (SqlConnection connection = new SqlConnection(connectionString))
                    {
                        connection.Open();
                        SqlCommand commandDeleteRow = new SqlCommand(queryDeleteRow, connection);
                        commandDeleteRow.Parameters.AddWithValue("@not", selectedNot);
                        commandDeleteRow.Parameters.AddWithValue("@atarih", selectedTarih);

                        commandDeleteRow.ExecuteNonQuery();
                        dataGridView1.Rows.Remove(selectedRow);
                    }
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Bir hata oluştu: {ex.Message}");
                }
            }
            else
            {
                MessageBox.Show("Lütfen silmek için bir satır seçin.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }
        }


        private void bunifuImageButton7_Click(object sender, EventArgs e)
        {
```

```csharp
            {
                //Veritabanına bağlı olan her noktayı güncelelyen kod.
                string kullaniciAdi = label16.Text;

                //Tablolardaki verileri güncelleeyerek alırken 2 sn liğine yükleme efekti veren kod.
                {
                    bunifuLoader1.Visible = true;
                    System.Timers.Timer timer = new System.Timers.Timer(1000); // 2 saniye
                    timer.Elapsed += (s, ev) =>
                    {
                        bunifuLoader1.Invoke(new Action(() => bunifuLoader1.Visible = false));
                        timer.Stop();
                        timer.Dispose();
                    };
                    timer.Start();
                }

                string connectionString = "Data Source=DESKTOP-242E7PH\\SQLEXPRESS;Initial Catalog=tbl_GelecegeNot;Integrated Security=True";

                try
                {
                    //İleriki tarihe ait notları veritabanından yeniden alıp sadece kullanıcıya ait Id ile uyuşan verileri tabloya yeniden getirir.
                    using (SqlConnection connection = new SqlConnection(connectionString))
                    {
                        string queryGetUserId = "SELECT id FROM kullanici WHERE kullaniciAdi = @kullaniciAdi";
                        SqlCommand command = new SqlCommand(queryGetUserId, connection);
                        command.Parameters.AddWithValue("@kullaniciAdi", kullaniciAdi);
                        connection.Open();
                        int kullaniciID = (int)command.ExecuteScalar();

                        string queryGetUserFutureNotes = "SELECT * FROM Gelecek WHERE kullaniciID = @kullaniciID ORDER BY atarih ASC";
                        SqlCommand command2 = new SqlCommand(queryGetUserFutureNotes, connection);
                        command2.Parameters.AddWithValue("@kullaniciID", kullaniciID);
                        SqlDataAdapter dataAdapter = new SqlDataAdapter(command2);
                        DataTable dataTable = new DataTable();
                        dataAdapter.Fill(dataTable);

                        dataGridView1.DataSource = dataTable;

                        //SQL sorgusu ile bugünün tarihini alıp veritabanında tarih stun)unda bugünün tarihi ile uyuşan notları Bugünün notları olan checkBox a taşır.
                        //Aynı zamanda SQL sorgusu ile bugünün tarihini alıp veritabanında tarih stununda bugünün tarihinden eski bir tarih var ise o satırı gelecek tablosundan silip geçmiş tablosuna yazdırır.
                        DateTime today = DateTime.Today;
                        string queryMoveData = @"
-- Gecmis tablosuna taşıma işlemi
INSERT INTO Gecmis ([not], tarih, kullaniciID)
SELECT [not], atarih AS tarih, kullaniciID
FROM Gelecek
WHERE atarih < @today;

-- Gelecek tablosundan taşınan verileri silme işlemi
DELETE FROM Gelecek
WHERE atarih < @today;
";
                        using (SqlCommand commandMoveData = new SqlCommand(queryMoveData, connection))
                        {
                            commandMoveData.Parameters.AddWithValue("@today", today);
                            commandMoveData.ExecuteNonQuery();
                        }
```

```csharp
                    }

                        string queryGetUserFutureNotes2 = "SELECT * FROM Gecmis WHERE kullaniciID = @kullaniciID ORDER BY tarih ASC";

                        SqlCommand command3 = new SqlCommand(queryGetUserFutureNotes2, connection);
                        command3.Parameters.AddWithValue("@kullaniciID", kullaniciID);
                        SqlDataAdapter dataAdapter2 = new SqlDataAdapter(command3);
                        DataTable dataTable2 = new DataTable();
                        dataAdapter2.Fill(dataTable2);
                        dataGridView2.DataSource = dataTable2;

                        string queryCheckFutureNotesToday = "SELECT [not] FROM Gelecek WHERE kullaniciID = @kullaniciID AND CONVERT(DATE, atarih) = @today";

                        SqlCommand commandCheckFutureNotesToday = new SqlCommand(queryCheckFutureNotesToday, connection);
                        commandCheckFutureNotesToday.Parameters.AddWithValue("@kullaniciID", kullaniciID);
                        commandCheckFutureNotesToday.Parameters.AddWithValue("@today", today);

                        SqlDataReader reader = commandCheckFutureNotesToday.ExecuteReader();
                        checkedListBox1.Items.Clear();

                        while (reader.Read())
                        {
                            string note = reader["not"].ToString();
                            checkedListBox1.Items.Add(note);
                        }
                        reader.Close();
                    }
                }
                catch (Exception ex)
                {
                    MessageBox.Show("Bir hata oluştu: " + ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }

            private void bunifuImageButton5_Click(object sender, EventArgs e)
            {
                //Veritabanına bağlanıp o kullanıcıya ait ID deki tüm notları gelecek tablosundan siler.
                DialogResult result = MessageBox.Show("Gelecek tablosundaki tüm notları silmek istediğinize emin misiniz?", "Uyarı", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

                if (result == DialogResult.Yes)
                {
                    string connectionString = "Data Source=DESKTOP-242E7PH\\SQLEXPRESS;Initial Catalog=tbl_GelecegeNot;Integrated Security=True";
                    string kullaniciAdi = label16.Text;

                    try
                    {
                        using (SqlConnection connection = new SqlConnection(connectionString))
                        {
                            string queryGetUserId = "SELECT id FROM kullanici WHERE kullaniciAdi = @kullaniciAdi";
                            SqlCommand commandGetUserId = new SqlCommand(queryGetUserId, connection);
                            commandGetUserId.Parameters.AddWithValue("@kullaniciAdi", kullaniciAdi);
                            connection.Open();

                            int kullaniciID = (int)commandGetUserId.ExecuteScalar();
                            string queryDeleteFutureNotes = "DELETE FROM gelecek WHERE kullaniciID = @kullaniciID";

                            SqlCommand commandDeleteFutureNotes = new SqlCommand(queryDeleteFutureNotes, connection);
                            commandDeleteFutureNotes.Parameters.AddWithValue("@kullaniciID", kullaniciID);
```

```csharp
                        commandDeleteFutureNotes.Parameters.AddWithValue("@kullaniciID", kullaniciID);
                        int rowsAffected = commandDeleteFutureNotes.ExecuteNonQuery();
                        connection.Close();
                        MessageBox.Show("Gelecek tablosundaki tüm notlar silindi.", "Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Information);
                    }
                }
                catch (Exception ex)
                {
                    MessageBox.Show("Bir hata oluştu: " + ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
        }


        private void bunifuImageButton3_Click(object sender, EventArgs e)
        {
            //Tablodaki seçili satırı tablodan ve veritabanından siler.
            if (dataGridView2.SelectedRows.Count > 0)
            {
                var selectedRow = dataGridView2.Cells["anot"].Value.ToString();

                var selectedNot = selectedRow.Cells["anot"].Value.ToString();
                var selectedTarih = Convert.ToDateTime(selectedRow.Cells["tarih"].Value);

                string connectionString = "Data Source=DESKTOP-242E7PH\\SQLEXPRESS;Initial Catalog=tbl_GelecegeNot;Integrated Security=True";
                string queryDeleteRow = "DELETE FROM Gecmis WHERE [not] = @not AND [tarih] = @tarih";
                try
                {
                    using (SqlConnection connection = new SqlConnection(connectionString))
                    {
                        connection.Open();
                        SqlCommand commandDeleteRow = new SqlCommand(queryDeleteRow, connection);
                        commandDeleteRow.Parameters.AddWithValue("@not", selectedNot);
                        commandDeleteRow.Parameters.AddWithValue("@tarih", selectedTarih);
                        commandDeleteRow.ExecuteNonQuery();
                        dataGridView2.Rows.Remove(selectedRow);
                    }
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Bir hata oluştu: {ex.Message}");
                }
            }
            else
            {
                MessageBox.Show("Lütfen silmek için bir satır seçin.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }
        }


        private void bunifuImageButton8_Click(object sender, EventArgs e)
        {
            //Mevcut formu kapatıp giriş ekranı formunu açar. Bu şekilde kullanıcı çıkış yapmış olur.
            Form2 FrmKlt = new Form2();
            FrmKlt.Show();
            this.Close();
        }
```

```csharp
        private void bunifuImageButton4_Click(object sender, EventArgs e)
        {
            //Geçmiş tablosundan ve veritabanından o kullanıcıya ait Id ile eş olan notları siler.
            DialogResult result = MessageBox.Show("Geçmiş tablosundaki tüm notları silmek istediğinize emin misiniz?", "Uyarı", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

            if (result == DialogResult.Yes)
            {
                // Bağlantı dizesi
                string connectionString = "Data Source=DESKTOP-242E7PH\\SQLEXPRESS;Initial Catalog=tbl_GelecegeNot;Integrated Security=True";

                string kullaniciAdi = label16.Text;

                try
                {
                    using (SqlConnection connection = new SqlConnection(connectionString))
                    {
                        string queryGetUserId = "SELECT id FROM kullanici WHERE kullaniciAdi = @kullaniciAdi";

                        SqlCommand commandGetUserId = new SqlCommand(queryGetUserId, connection);
                        commandGetUserId.Parameters.AddWithValue("@kullaniciAdi", kullaniciAdi);

                        connection.Open();
                        int kullaniciID = (int)commandGetUserId.ExecuteScalar();

                        string queryDeleteFutureNotes = "DELETE FROM gecmis WHERE kullaniciID = @kullaniciID";

                        SqlCommand commandDeleteFutureNotes = new SqlCommand(queryDeleteFutureNotes, connection);
                        commandDeleteFutureNotes.Parameters.AddWithValue("@kullaniciID", kullaniciID);

                        int rowsAffected = commandDeleteFutureNotes.ExecuteNonQuery();
                        connection.Close();
                        MessageBox.Show("Geçmiş tablosundaki tüm notlar silindi.", "Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Information);
                    }
                }
                catch (Exception ex)
                {
                    MessageBox.Show("Bir hata oluştu: " + ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
        }

        private void bunifuImageButton6_Click(object sender, EventArgs e)
        {
            //Seçenekler kısmını görünür yada gizli ayarlayan kod.
            bunifuShadowPanel1.Visible = !bunifuShadowPanel1.Visible;
        }

        private void bunifuImageButton12_Click(object sender, EventArgs e)
        {
            //Kullanıcının şifresini yenileyebileceği yer. Kullanıcı adından o ID yi bulur ardından o id ye ait satırdaki şifre sütununu textBox a girilen veri ile değiştirir.
            string kullaniciAdiGuncellenecek = label16.Text;
            string yeniKullaniciAdi = bunifuTextBox1.Text;

            if (string.IsNullOrWhiteSpace(yeniKullaniciAdi))
            {
                MessageBox.Show("Lütfen yeni bir kullanıcı adı girin.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
```

```csharp
                    return;
                }
                string connectionString = "Data Source=DESKTOP-242E7PH\\SQLEXPRESS;Initial Catalog=tbl_GelecegeNot;Integrated Security=True";
                string query = "UPDATE kullanici SET kullaniciAdi = @yeniKullaniciAdi WHERE kullaniciAdi = @kullaniciAdiGuncellenecek";

                using (SqlConnection connection = new SqlConnection(connectionString))
                {
                    SqlCommand command = new SqlCommand(query, connection);
                    command.Parameters.AddWithValue("@yeniKullaniciAdi", yeniKullaniciAdi);
                    command.Parameters.AddWithValue("@kullaniciAdiGuncellenecek", kullaniciAdiGuncellenecek);

                    try
                    {
                        connection.Open();
                        int rowsAffected = command.ExecuteNonQuery();
                        if (rowsAffected > 0)
                        {
                            MessageBox.Show("Kullanıcı adı başarıyla güncellendi.", "Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Information);
                            label16.Text = yeniKullaniciAdi;
                            bunifuTextBox1.Text = null;
                        }
                        else
                        {
                            MessageBox.Show("Güncelleme işlemi başarısız oldu. Lütfen tekrar deneyin.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
                        }
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show("Bir hata oluştu: " + ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
                    }
                    finally
                    {
                        connection.Close();
                    }
                }
            }

            private void bunifuImageButton11_Click(object sender, EventArgs e)
            {
                //Kullanıcının Kullanıcı adını yenileyebileceği yer. Kullanıcı adından ID yi bulur ardından o ıd ye ait satırdaki kullaniciAdi stünunu textBox a girilen veri ile değiştirir.
                string kullaniciAdi = label16.Text;
                string yeniSifre = bunifuTextBox2.Text;
                string connectionString = "Data Source=DESKTOP-242E7PH\\SQLEXPRESS;Initial Catalog=tbl_GelecegeNot;Integrated Security=True";
                string query = "UPDATE kullanici SET sifre = @yeniSifre WHERE kullaniciAdi = @kullaniciAdi";

                using (SqlConnection connection = new SqlConnection(connectionString))
                {
                    SqlCommand command = new SqlCommand(query, connection);
                    command.Parameters.AddWithValue("@yeniSifre", yeniSifre);
                    command.Parameters.AddWithValue("@kullaniciAdi", kullaniciAdi);

                    try
                    {
                        connection.Open();
                        int rowsAffected = command.ExecuteNonQuery();

                        if (rowsAffected > 0)
                        {
                            MessageBox.Show("Şifre başarıyla güncellendi.", "Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Information);
                            bunifuTextBox2.Text = null;
                        }
                        else
                        {
                            MessageBox.Show("Güncelleme işlemi başarısız oldu. Lütfen tekrar deneyin.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
                        }
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show("Bir hata oluştu: " + ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
                    }
                    finally
                    {
                        connection.Close();
                    }
                }
            }

            private void bunifuImageButton9_Click(object sender, EventArgs e)
            {
                //Kullanıcıya ait her şeyi siler. Kullanıcının ID sini alır ve gelecek , geçmiş ve kullanıcı tablolarından o ıd ye ait satırları silen kod.
                string kullaniciAdi = label16.Text;
                DialogResult result = MessageBox.Show("Kullanıcıyı ve kullanıcıya ait kayıtları silmek istediğinizden emin misiniz?", "Onay", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

                if (result == DialogResult.Yes)
                {
                    string connectionString = "Data Source=DESKTOP-242E7PH\\SQLEXPRESS;Initial Catalog=tbl_GelecegeNot;Integrated Security=True";

                    try
                    {
                        using (SqlConnection connection = new SqlConnection(connectionString))
                        {
                            connection.Open();
                            string queryGetUserId = "SELECT id FROM kullanici WHERE kullaniciAdi = @kullaniciAdi";
                            SqlCommand commandGetUserId = new SqlCommand(queryGetUserId, connection);
                            commandGetUserId.Parameters.AddWithValue("@kullaniciAdi", kullaniciAdi);
                            object resultUserId = commandGetUserId.ExecuteScalar();
                            int kullaniciID = -1;
                            if (resultUserId != null)
                            {
                                kullaniciID = Convert.ToInt32(resultUserId);
                            }

                            if (kullaniciID != -1)
                            {
                                string queryDeleteFromGelecek = "DELETE FROM gelecek WHERE kullaniciID = @kullaniciID";
                                string queryDeleteFromKullanici = "DELETE FROM kullanici WHERE id = @kullaniciID";
                                SqlCommand commandDeleteFromGelecek = new SqlCommand(queryDeleteFromGelecek, connection);
                                commandDeleteFromGelecek.Parameters.AddWithValue("@kullaniciID", kullaniciID);

                                SqlCommand commandDeleteFromKullanici = new SqlCommand(queryDeleteFromKullanici, connection);
                                commandDeleteFromKullanici.Parameters.AddWithValue("@kullaniciID", kullaniciID);
                                commandDeleteFromGelecek.ExecuteNonQuery();
                                commandDeleteFromKullanici.ExecuteNonQuery();

                                MessageBox.Show("Kullanıcı ve kullanıcıya ait kayıtlar başarıyla kaldırıldı.", "Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Information);
```

```csharp
                                MessageBox.Show("Kullanıcı ve kullanıcıya ait kayıtlar başarıyla kaldırıldı.", "Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Information);
                                Form2 FrmKlt = new Form2();
                                FrmKlt.Show();
                                this.Close();
                            }
                            else
                            {
                                MessageBox.Show("Kullanıcı bulunamadı.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
                            }
                        }
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show($"Bir hata oluştu: {ex.Message}");
                    }
                }
            }

            private void bunifuImageButton2_Click(object sender, EventArgs e)
            {
                //Notu düzenlemeyi sağlar. SEçili satırı tabodan ve veritabanından siler ve not ve tarihi not ekleme kısmına a yzdırır.
                if (dataGridView1.SelectedRows.Count > 0)
                {
                    var selectedRow = dataGridView1.SelectedRows[0];
                    var selectedNot = selectedRow.Cells["not"].Value.ToString();
                    var selectedTarih = Convert.ToDateTime(selectedRow.Cells["atarih"].Value);

                    TxtBoxNOT.Text = selectedNot;
                    TxtBoxTARİH.Text = selectedTarih.ToString("yyyy-MM-dd");
                    string connectionString = "Data Source=DESKTOP-242E7PM\\SQLEXPRESS;Initial Catalog=tbl_GelecegeNot;Integrated Security=True";
                    string queryDeleteRow = "DELETE FROM Gelecek WHERE [not] = @not AND [atarih] = @atarih";

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

                            SqlCommand commandDeleteRow = new SqlCommand(queryDeleteRow, connection);
                            commandDeleteRow.Parameters.AddWithValue("@not", selectedNot);
                            commandDeleteRow.Parameters.AddWithValue("@atarih", selectedTarih);
                            commandDeleteRow.ExecuteNonQuery();
                            dataGridView1.Rows.Remove(selectedRow);
                        }
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show($"Bir hata oluştu: {ex.Message}", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
                    }
                }
                else
                {
                    MessageBox.Show("Değiştirmek istedeğiniz satırı seçin.", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                }
            }
        }
    }
```