# 

T.C.

CUMHURİYET ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

# Arkadaş öneri sistemi

## Öğrenci Adı: BATUHAN ÇAM

## Numarası: 2015141057

### *Özet:*

Bu projede özetle elimizde bilgileri bulunan 90 tane öğrencinin ve bu öğrencilerin belirlenen 15 tane soruya verdiği 0 ile 10 arasındaki cevaplar bulunmaktadır.Ve bu bilgilere dayanarak numarası girilen kişiye arkadaş önerisinde bulunulacaktır.Arkadaş önerilirken lojistik regresyon algoritması kullanılacaktır

***Anahtar Kelimeler:*** Arkadaş,öneri,lojistik regresyon

*Abstract*

In this project, there are 90 students who have information in detail and 0 to 10 answers given by these students. Based on this information, friend suggestions will be made to the entered person. Logistic regression algorithm will be used when a friend is suggested

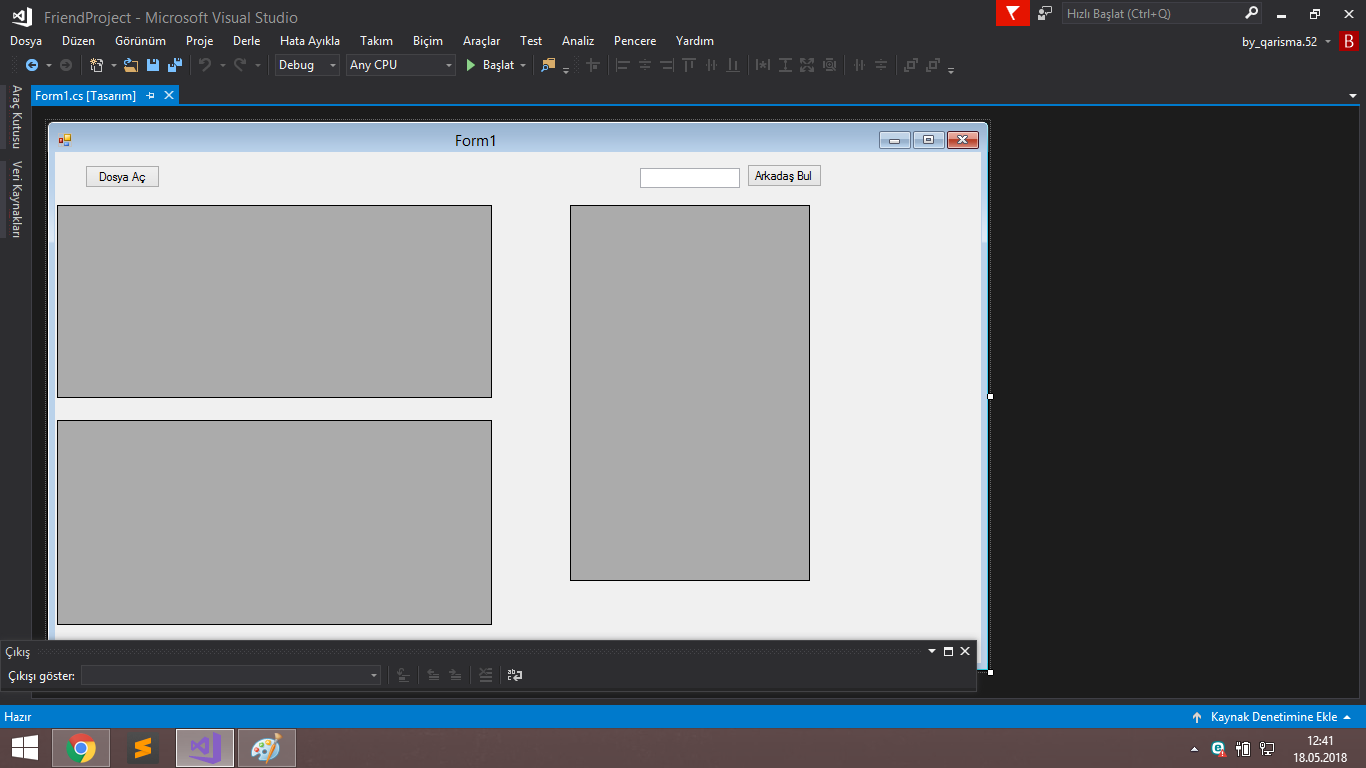
***Keywords:*** Friend, suggestion , logistic regression

## Giriş

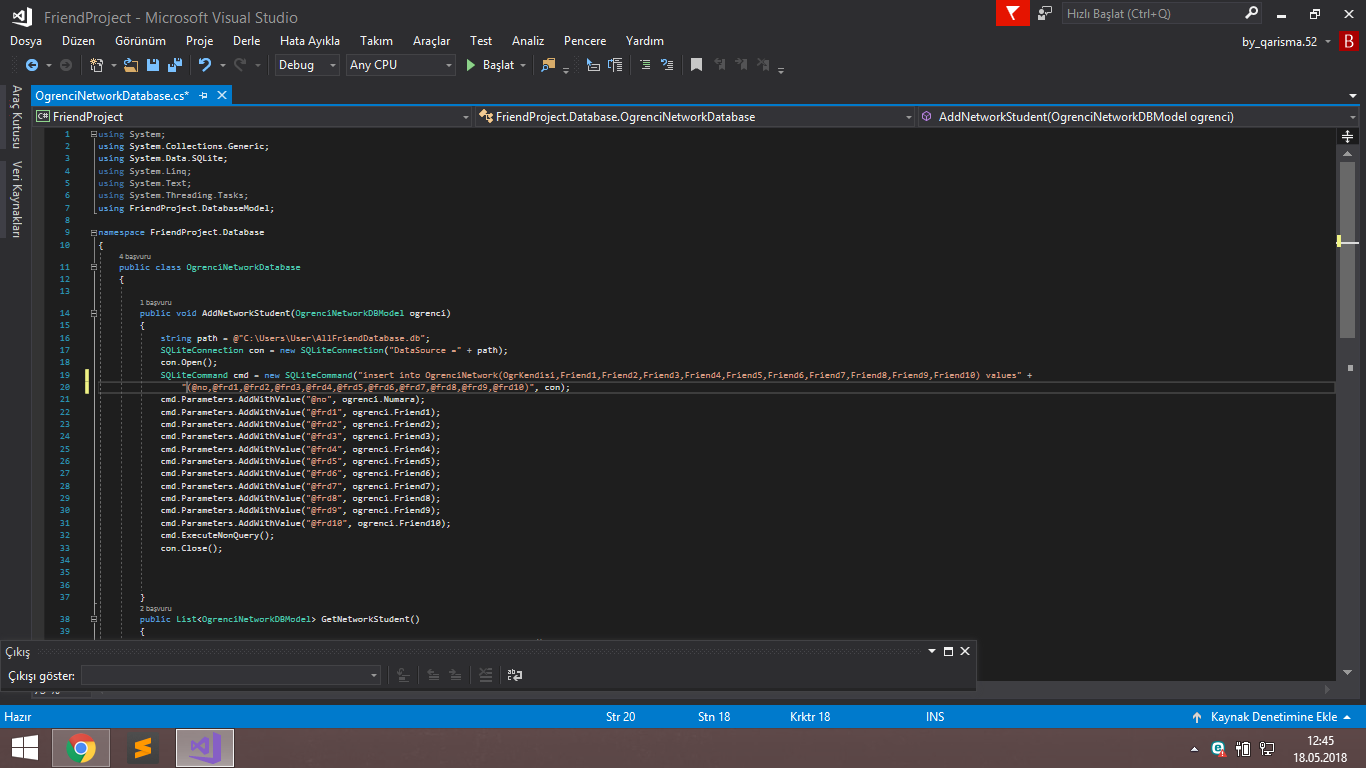
Bu projede elimizde 90 tane öğrenci ve bu öğrencilerin en yakın arkadaşları ve aynı zamanda her bir öğrencinin anket sorularına verdiği 0 ile 10 arasında puanların bulunduğu iki adet dosya bulunmaktadır.Bizim amacımız ve problemimiz bu dosyalardaki bilgileri kullanarak numarası girilen kişiye 10 adet arkadaş önerisinde bulunmaktır. Kullanıcı arayüzünde dosyaların okunacağı bir dosya seçme butonu olacak ve bu seçilen dosyaların içeriği gösterilecektir. Sonrasında bir textbox yardımı ile kullanıcı sistemin arkadaş önermesini istediği numarayı girecek ve numarayı girdikten sonra “Arkadaş Bul” butonuna basınca önerilen 10 tane arkadaş GridView de gösterilecektir. Öğrencilerin arkadaş ve profil bilgilerine göre arkadaş önerebilmek için lojistik regresyon algoritması kullanılmalıdır.

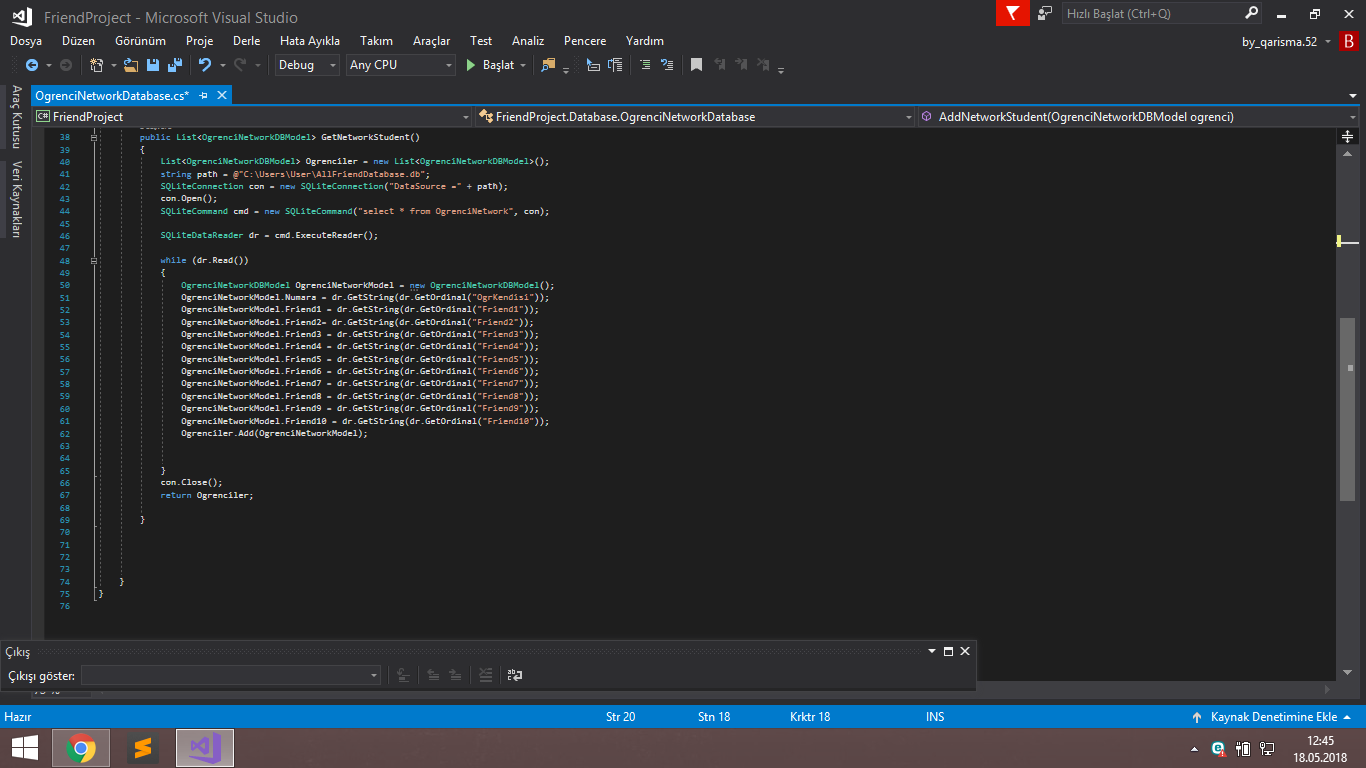
## UYGULAMA

1-) User Interface : Uygulamanın kullanıcı arayüzü kullanıcının işlem yapacağı kısım 2 adet buton ve 3 adet grid view var.Buton işleyişleri çıktı kısmında gösterilecektir.

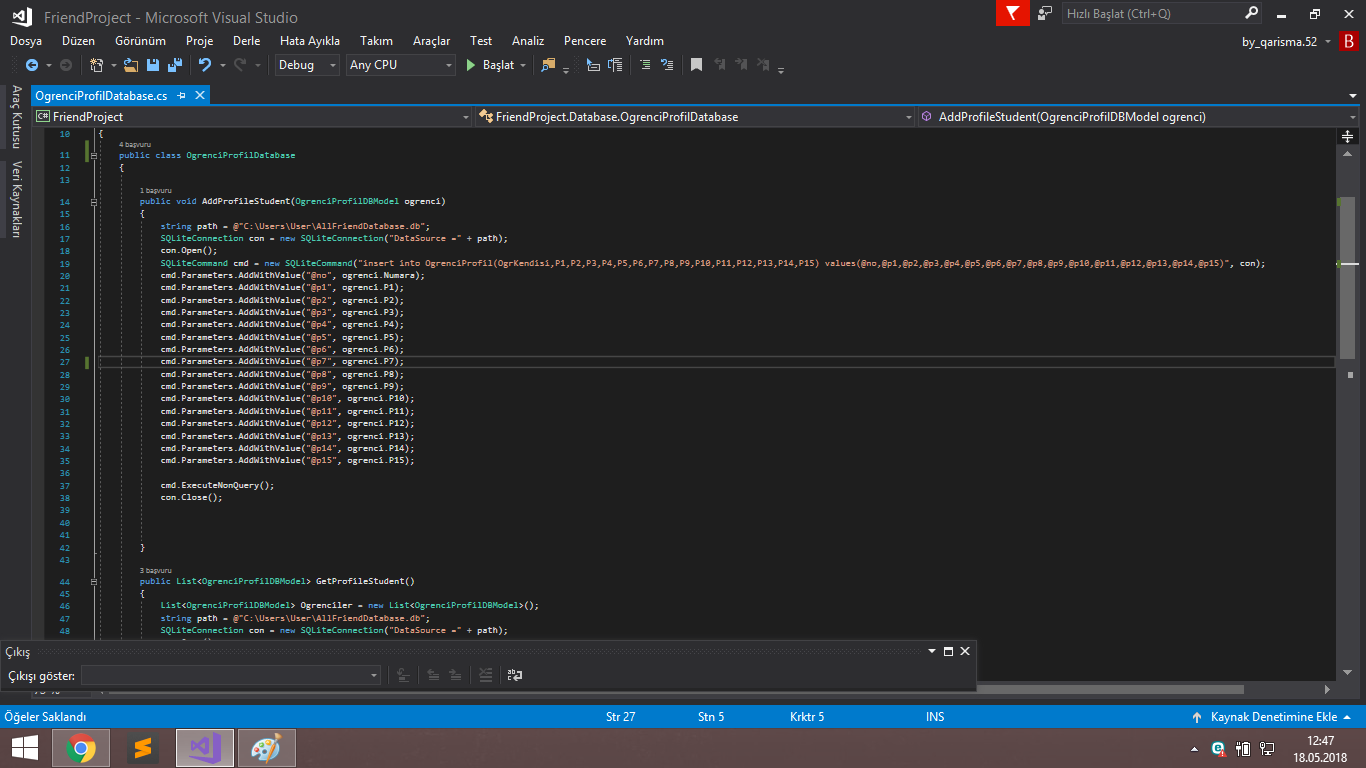


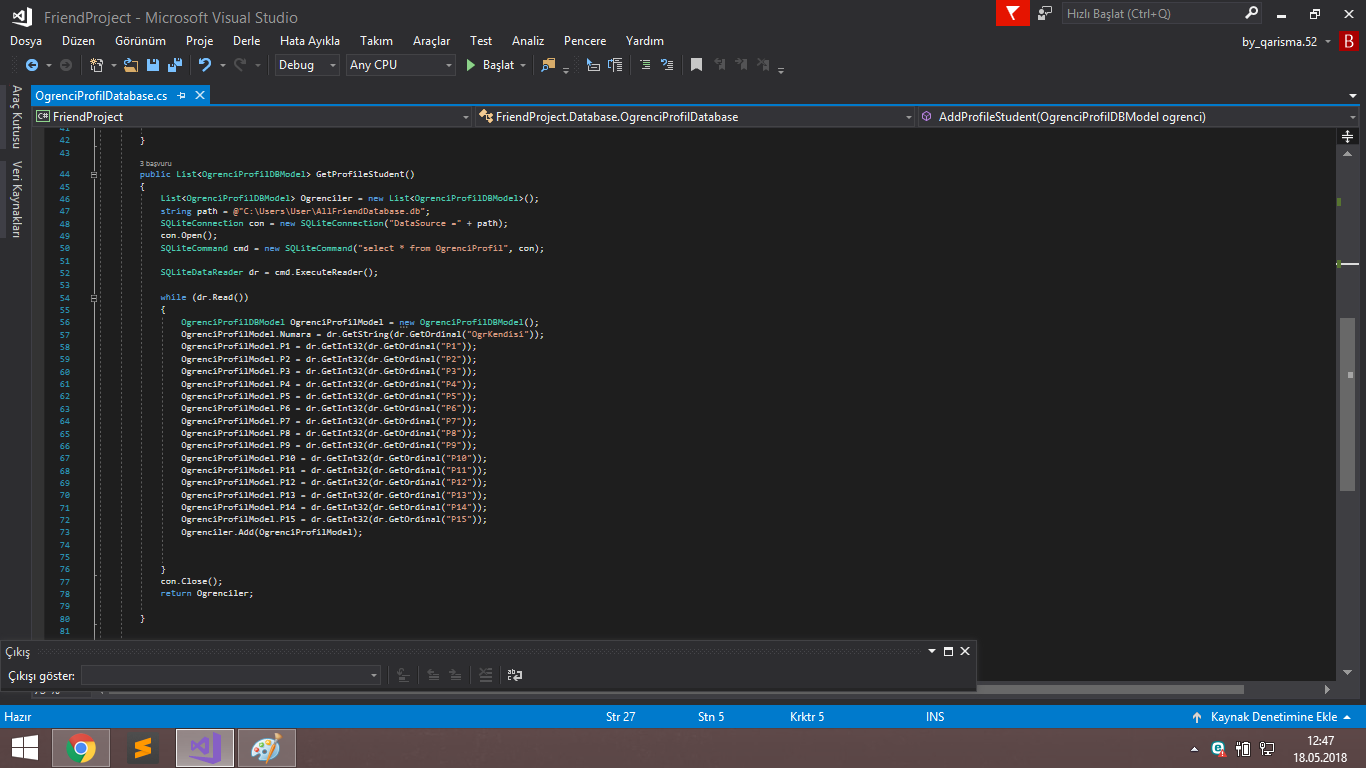
2-)OgrenciNetworkDatabase Sınıfı : Bu sınıfta OgrenciNetwork tablosuna dosyadan okunacak olan verileri eklemek ve eklenen verileri tablodan geri getirmek için gereken fonksiyonlar yazılmıştır.



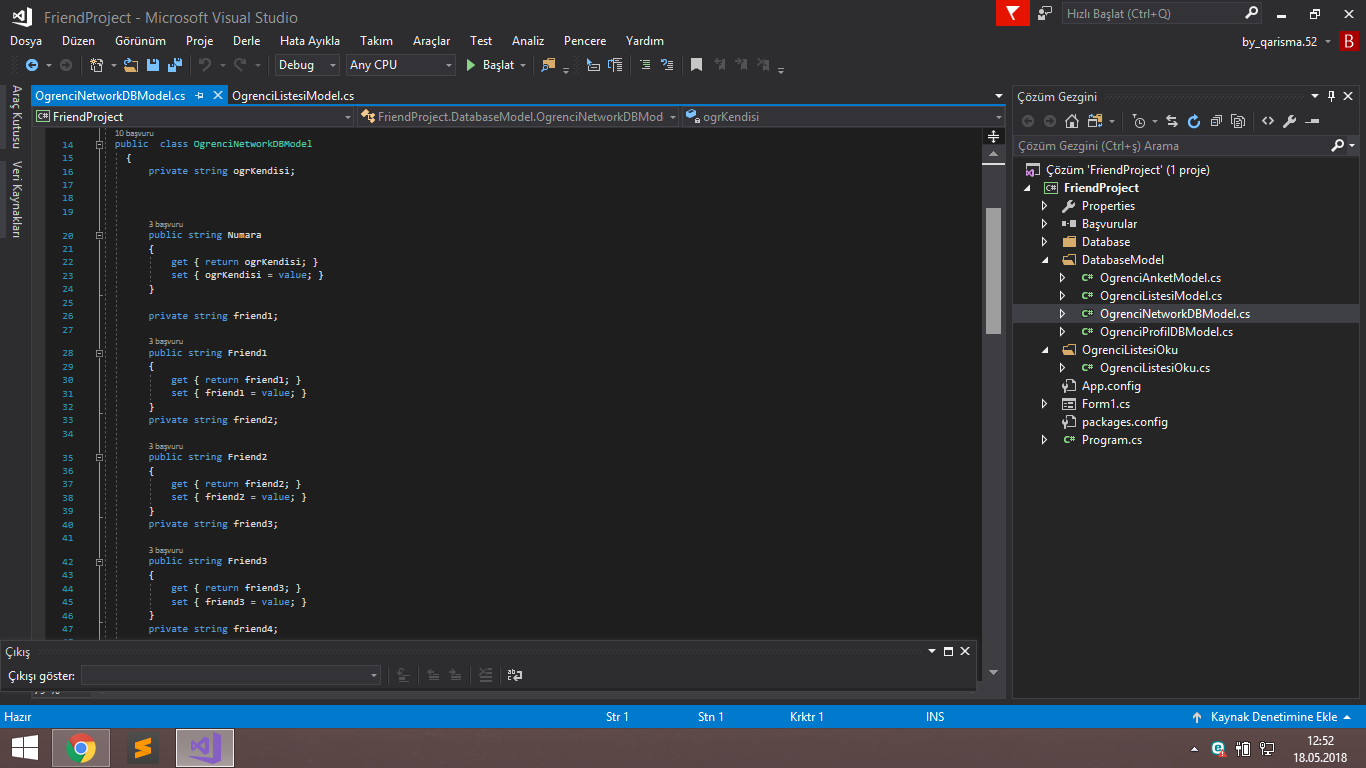


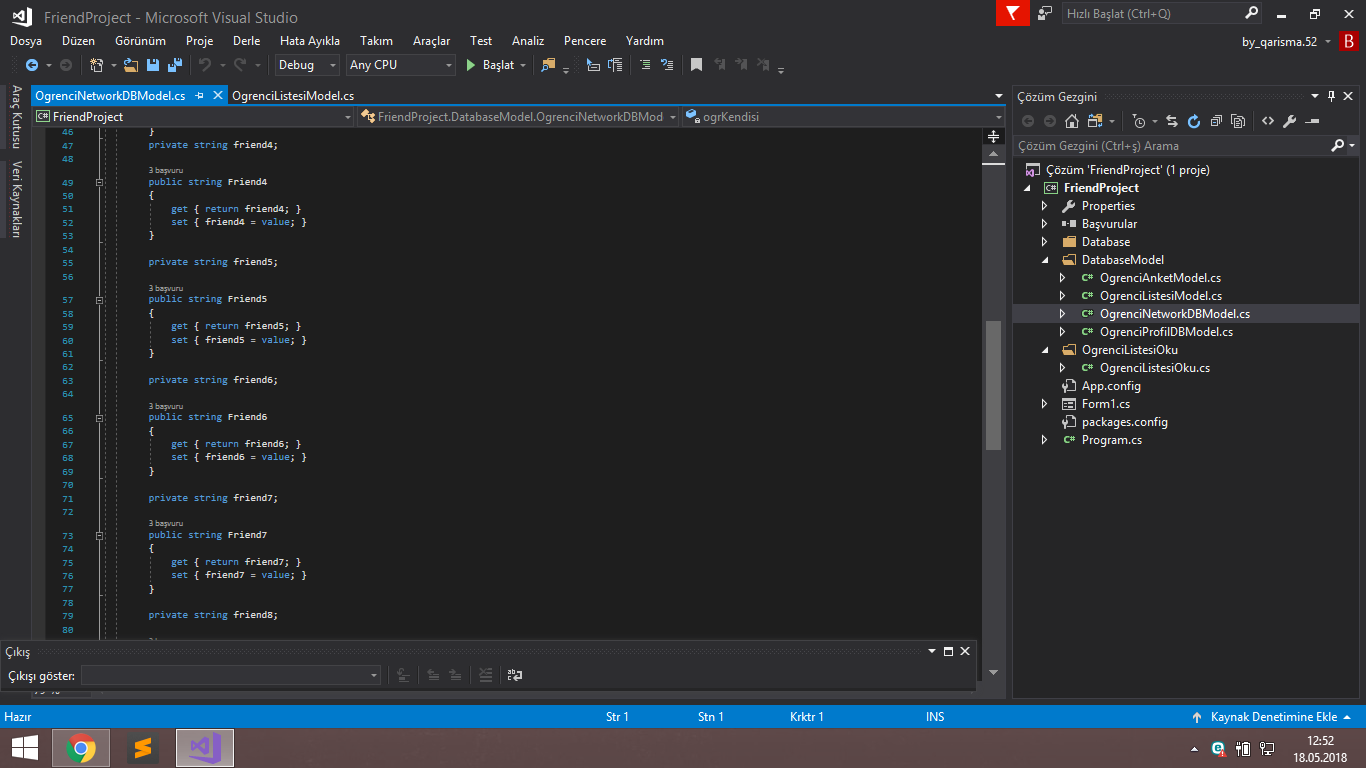
3-)OgrenciProfilDatabase Sınıfı : Bu sınıftada network sınıfında olduğu gibi dosyadan okunacak olan verilerin OgrenciProfil tablosuna eklenmesi ve ardından geri getirilmesi için gereken fonksiyonlar yazılmıştır.

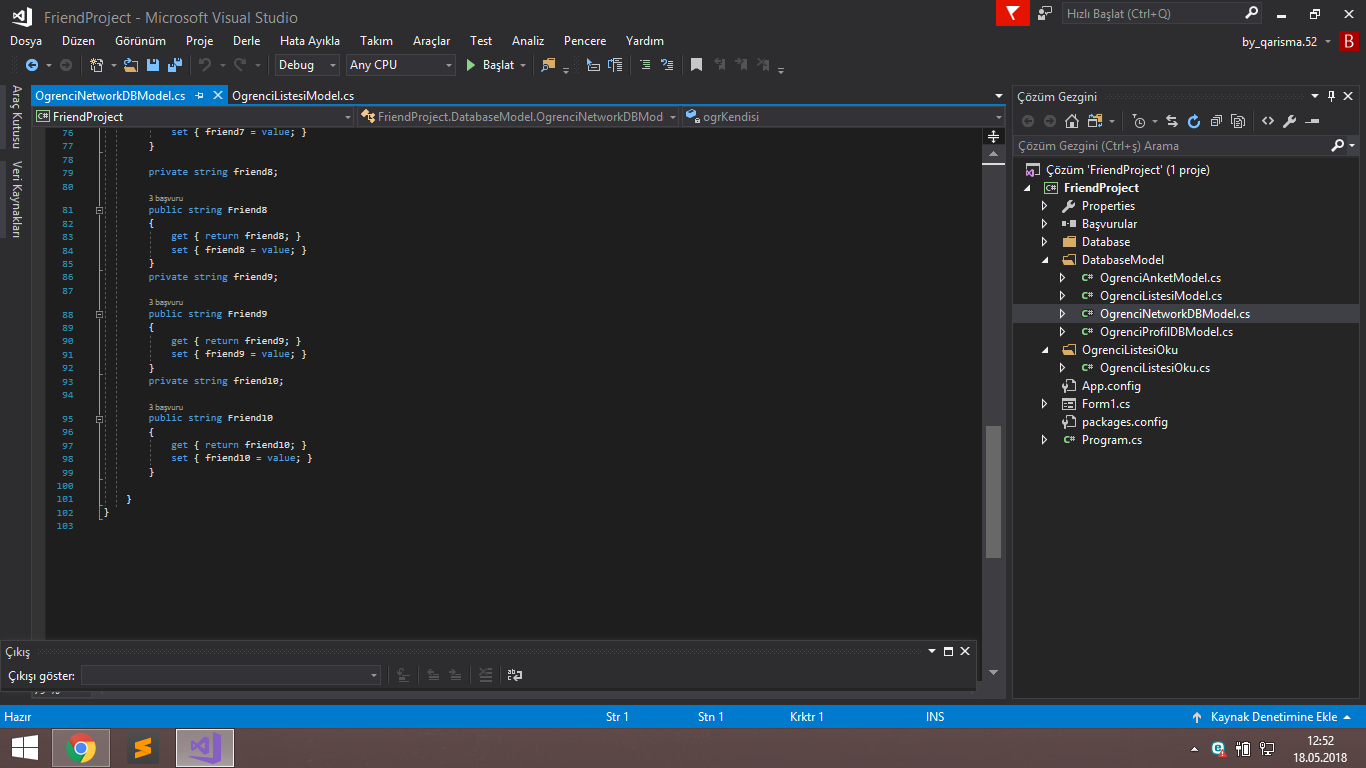




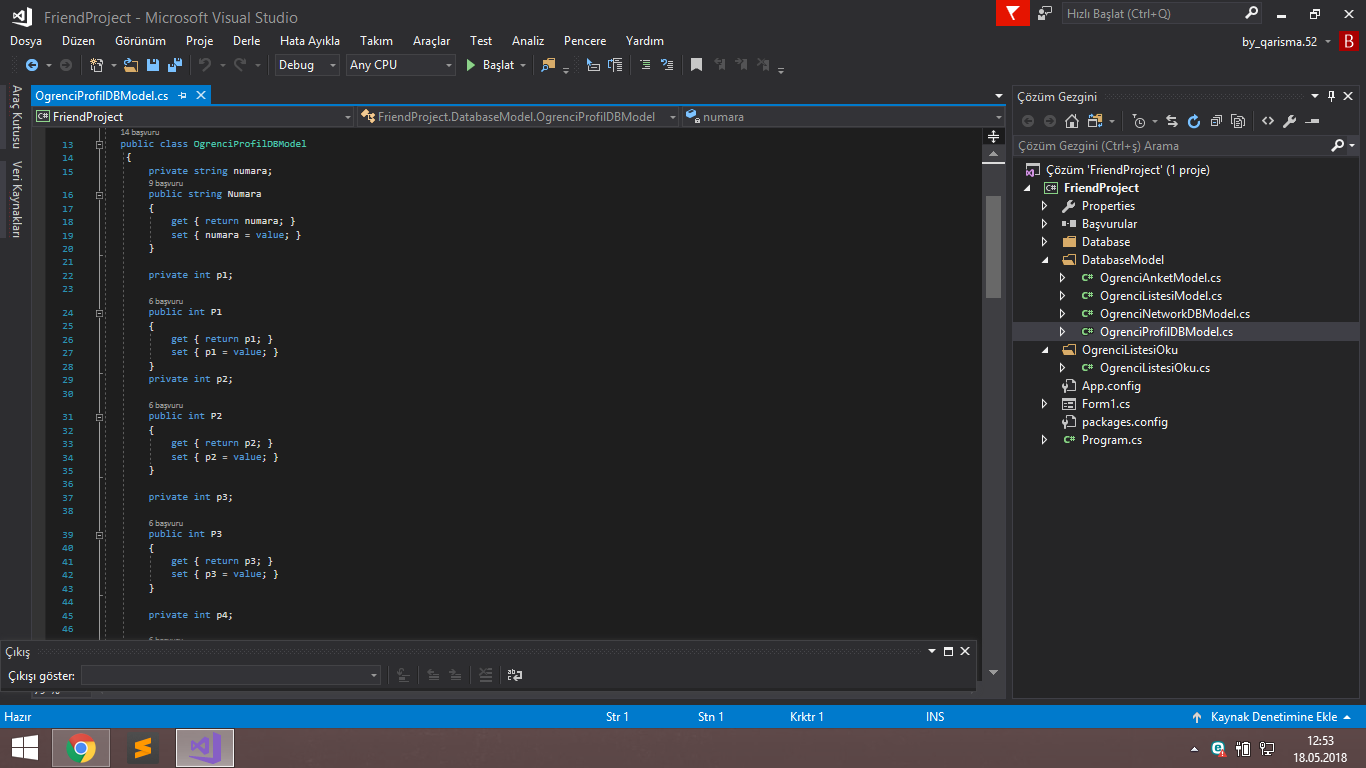
4-)OgrenciNetworkDBModel Sınıfı : Bu sınıf OgrenciNetwork tablomuzun modelidir yani kodun içerisinde tablo sütunları burda tutulur datalar bu model sayesinde veritabanına kayıt olur.

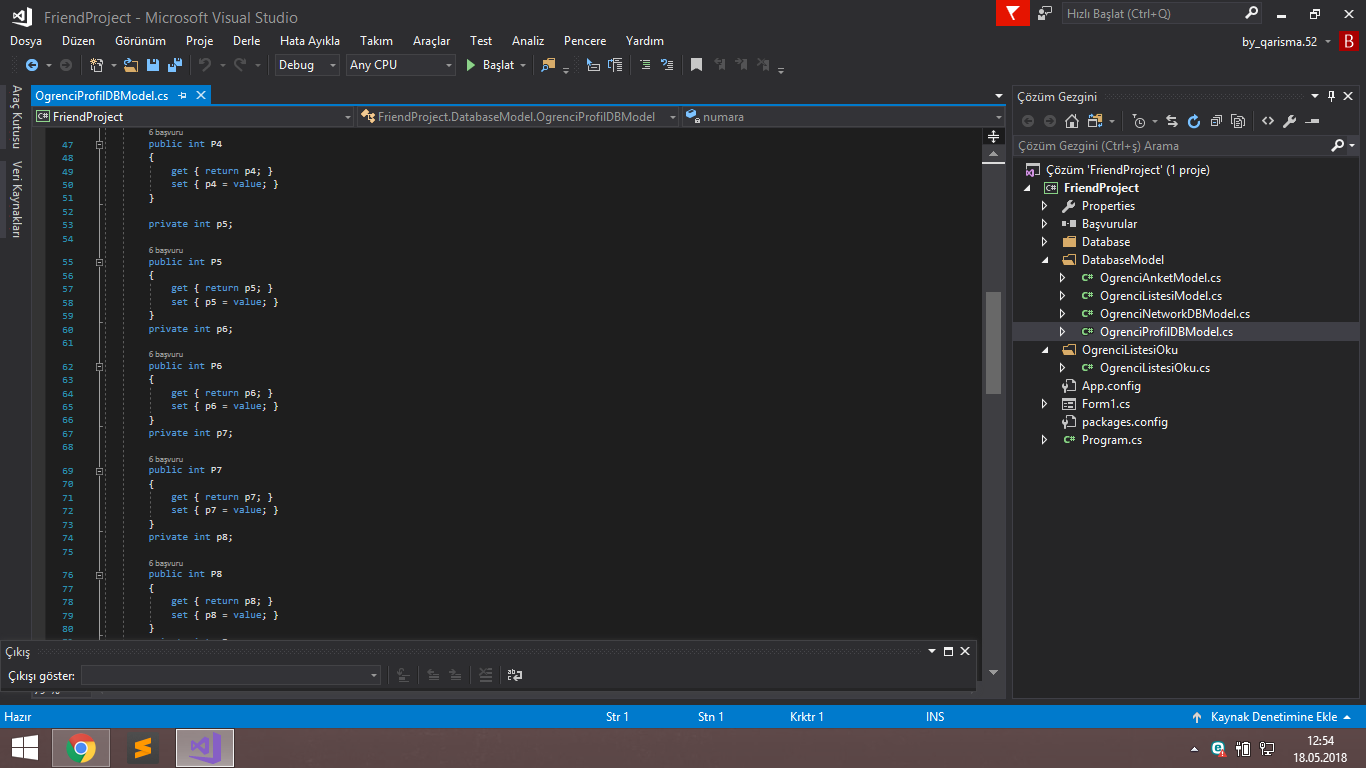


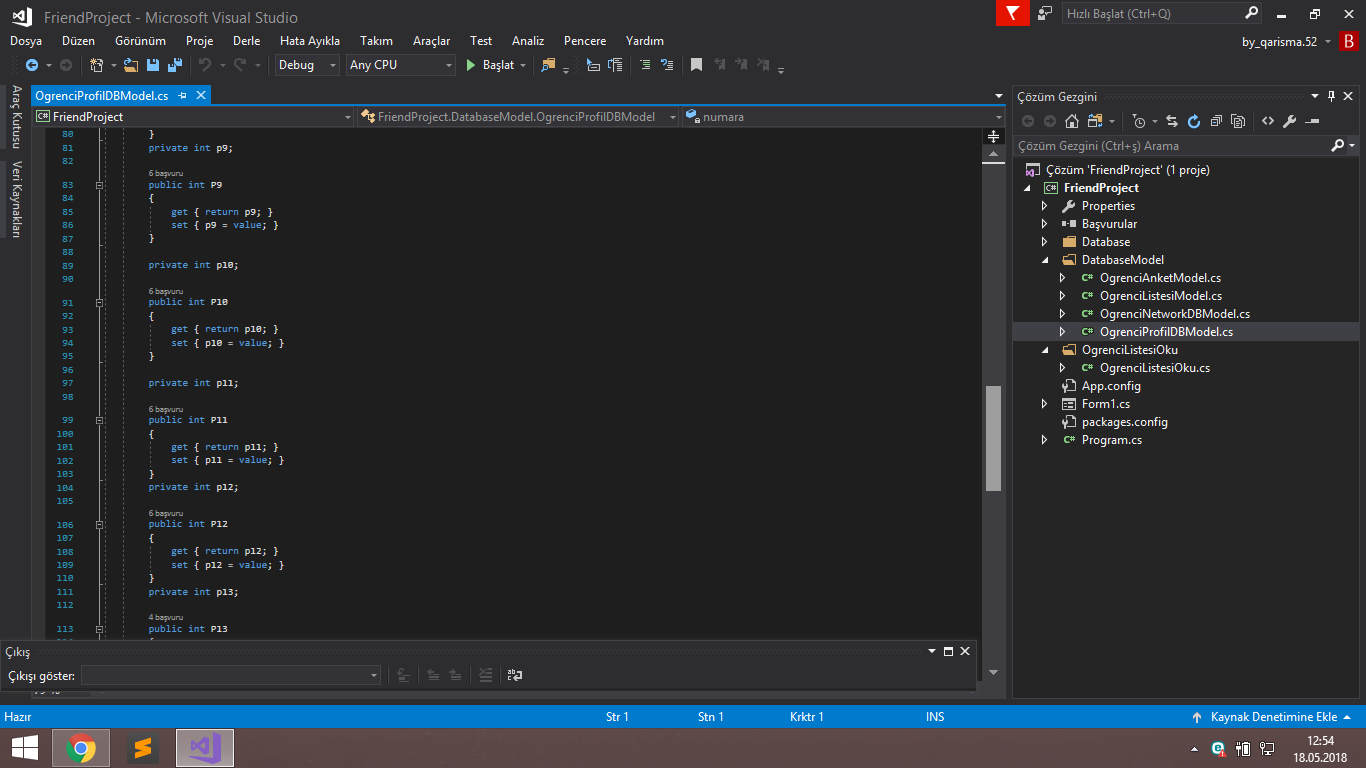


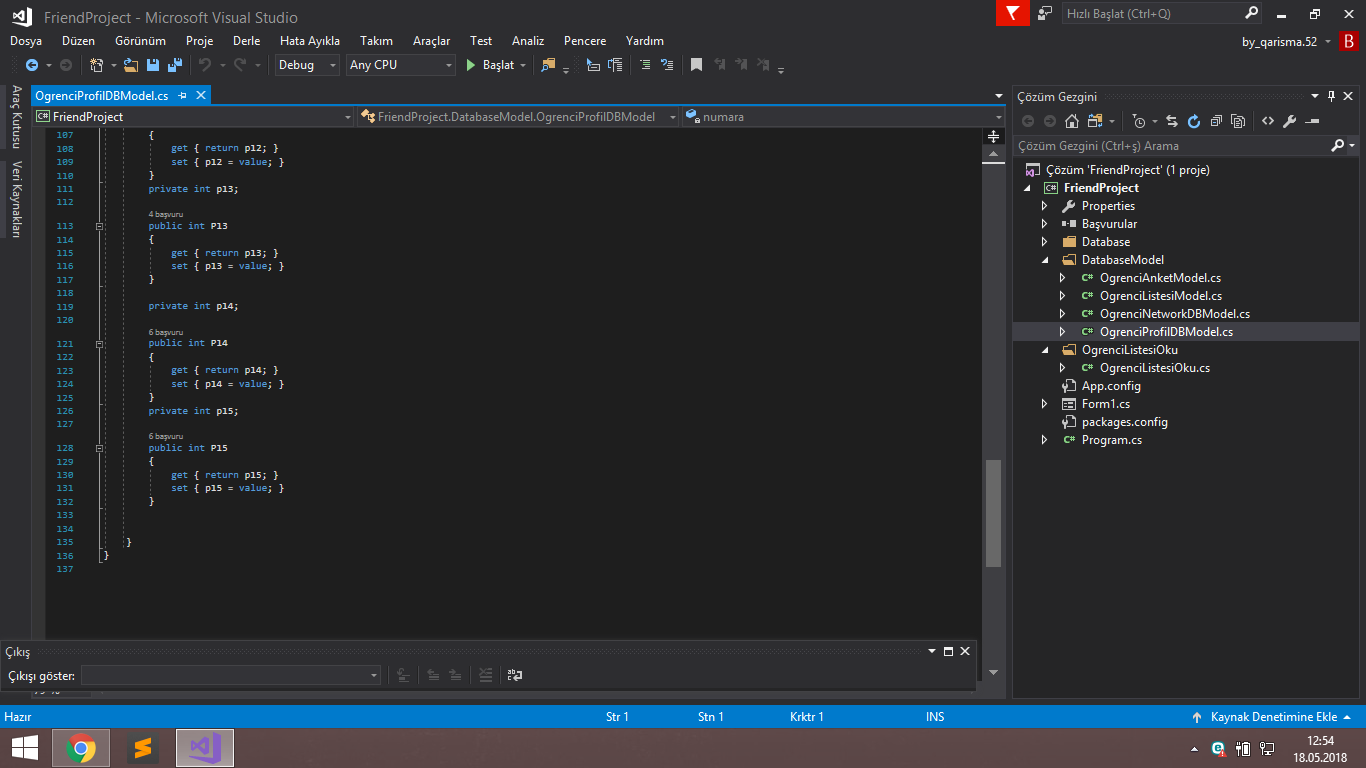


5-)OgrenciProfilDBModel : Bu modelde aynı şekilde profil bilgilerini kaydetmek için kullanılır.

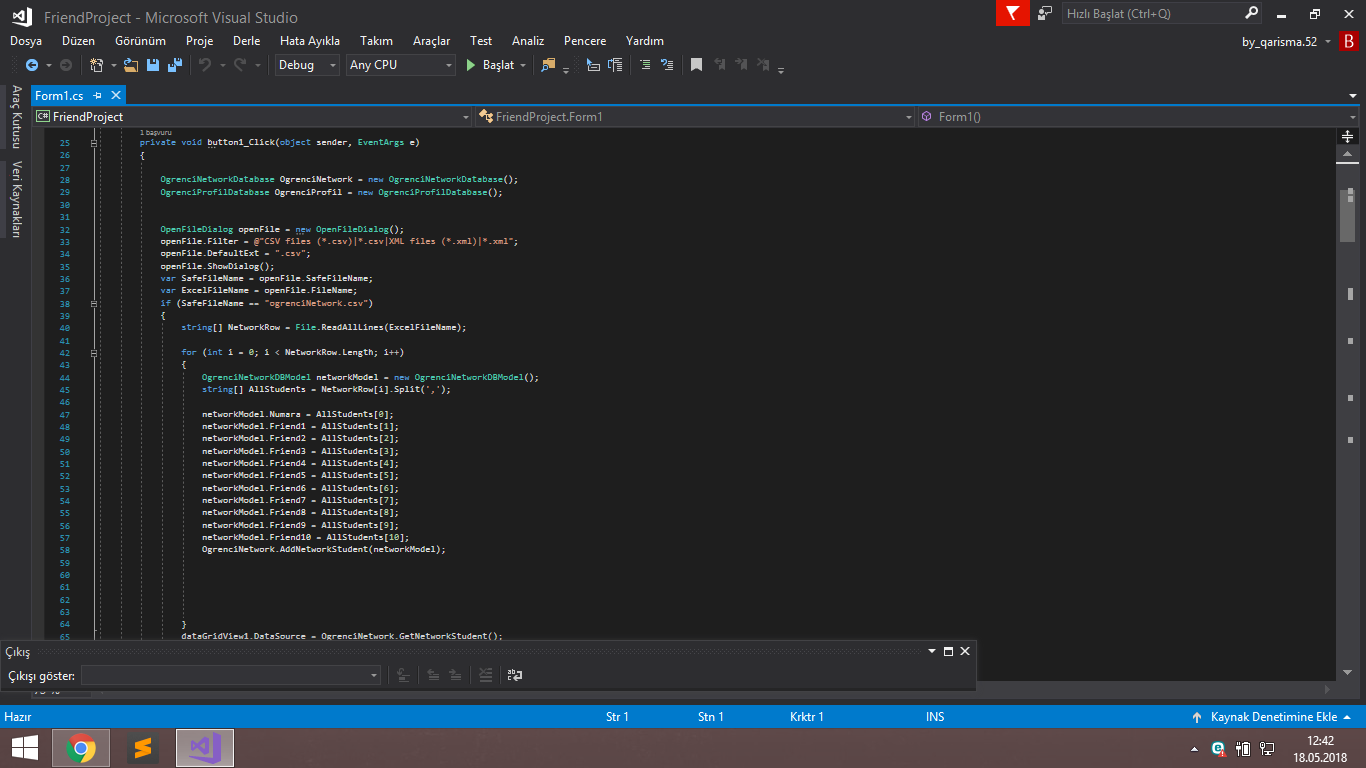


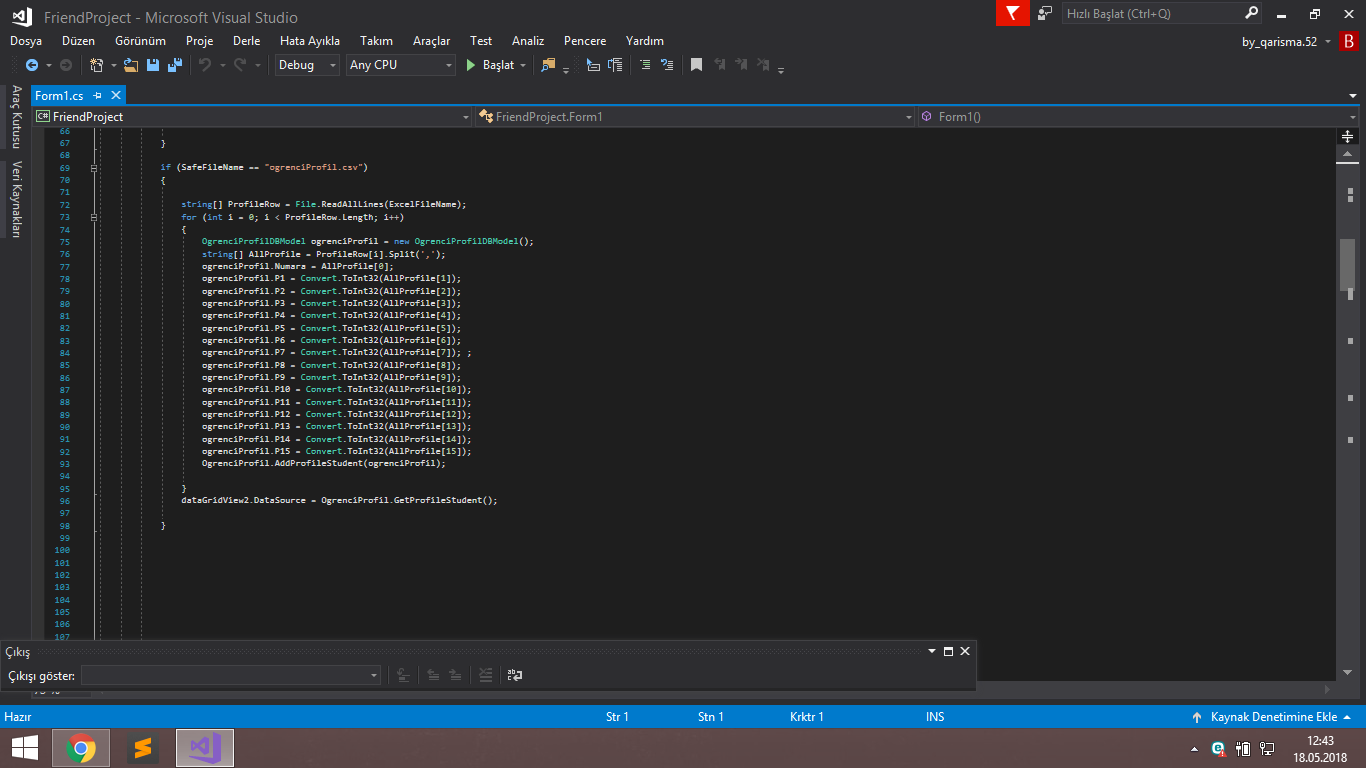




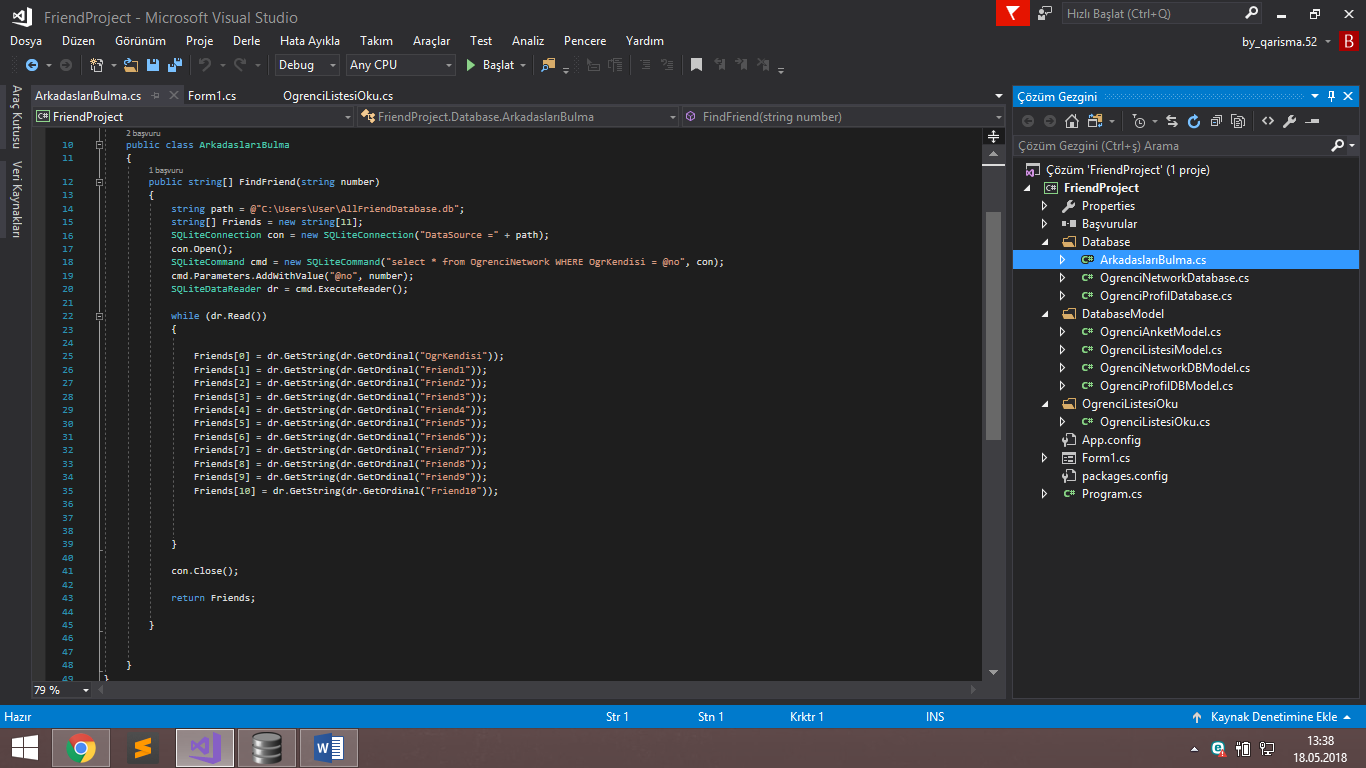


6-)Button1 Click Fonksiyonu : Bu fonksiyonda elimizdeki ogrenciProfil ve ogrenciNetwork excel dosyalarını bir open file dialog yardımıyla seçtirip ardından seçilen dosyayı okuyup ilgili tabloya kaydedip ardında gridview de göstermek için bu kaydettiğimiz verileri geri getirip gridviewlere basıyoruz.

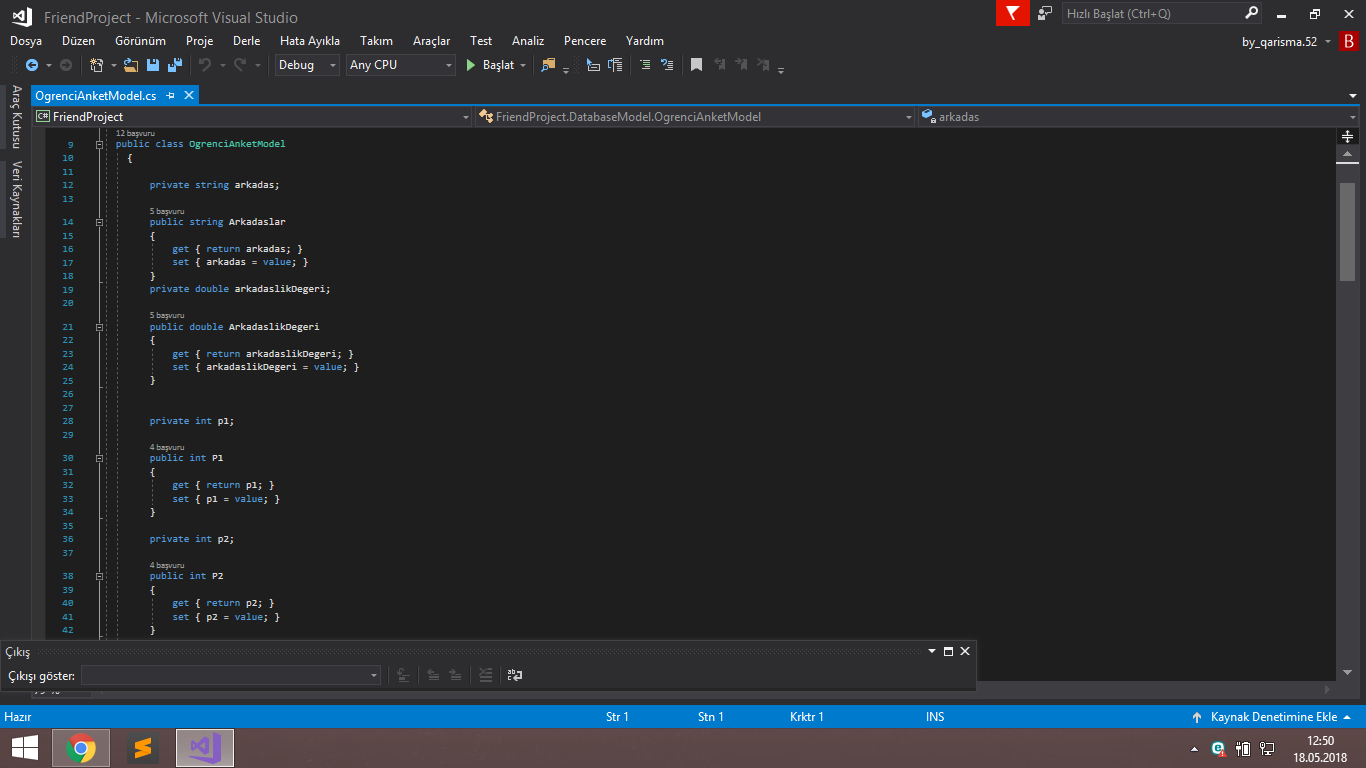




## 7-)Arkadasbulma sınıfı :Bu sınıfta girilen numaraya göre veritabanından o kişi ve arkadaşları alınır.



8-)OgrenciAnketModel Sınıfı : Bu sınıf logistik regresyon için gerekli model sınıfıdır. Regresyon için gerekli olan değişkenler burada tutulur.

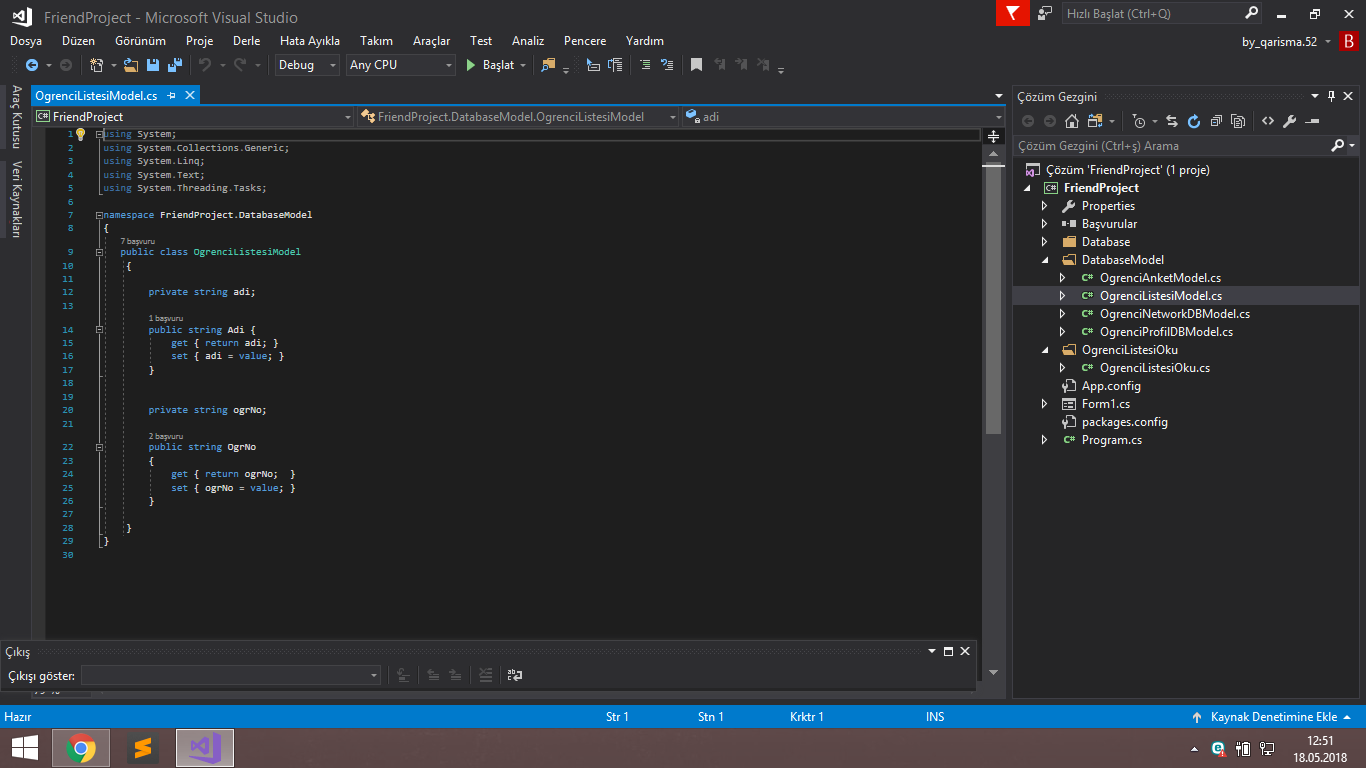


## 

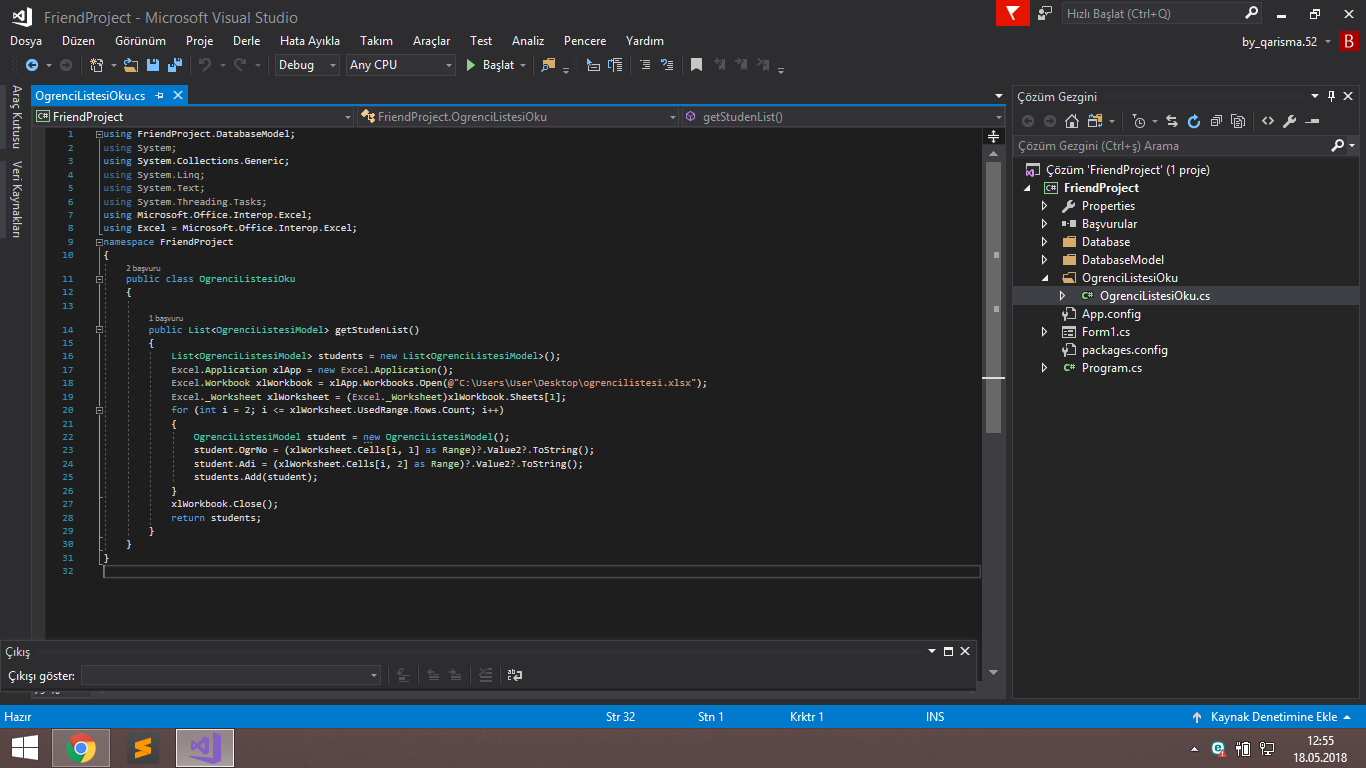
## 

## 

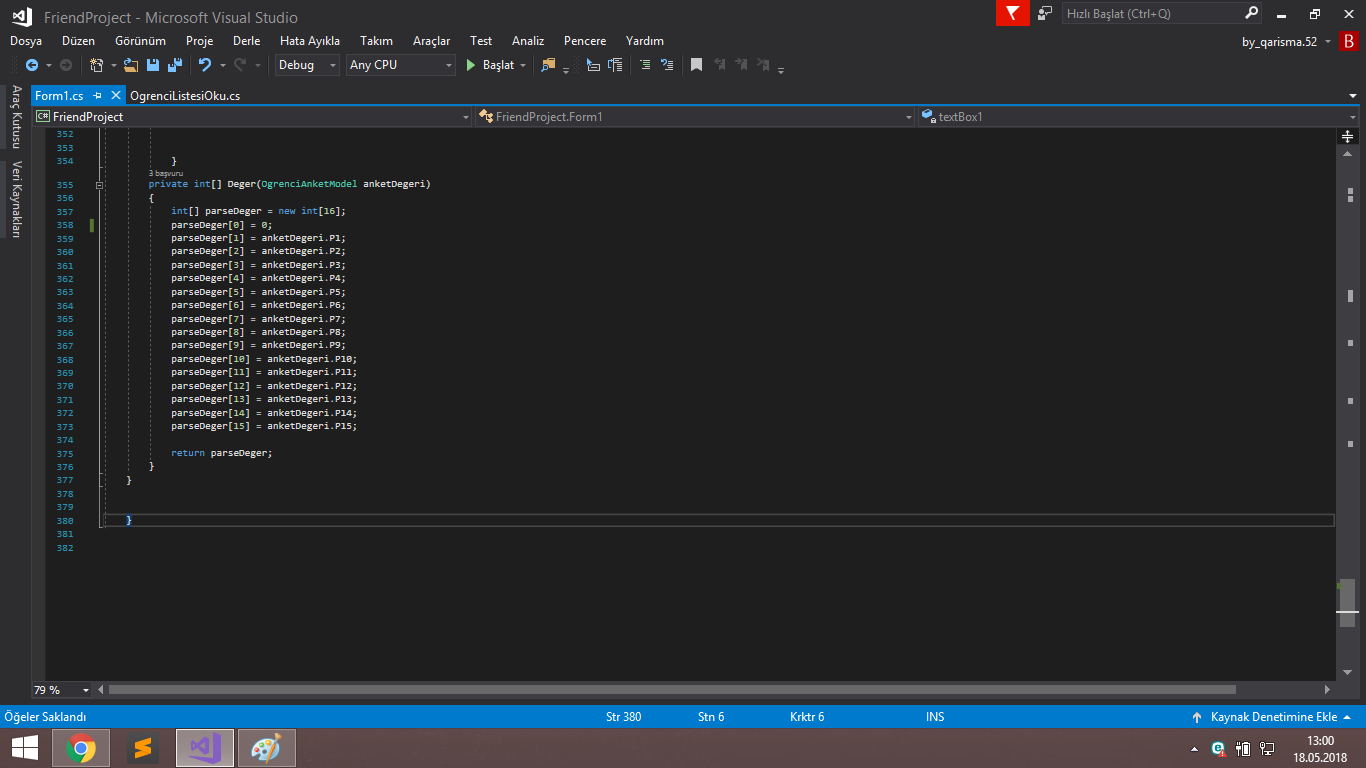
9-) OgrenciListesiModel Sınıfı : Bu sınıfta önerilen öğrencilerin sadece numarasını değil adını da göstermek için gerekli değişkenler tanımlı.



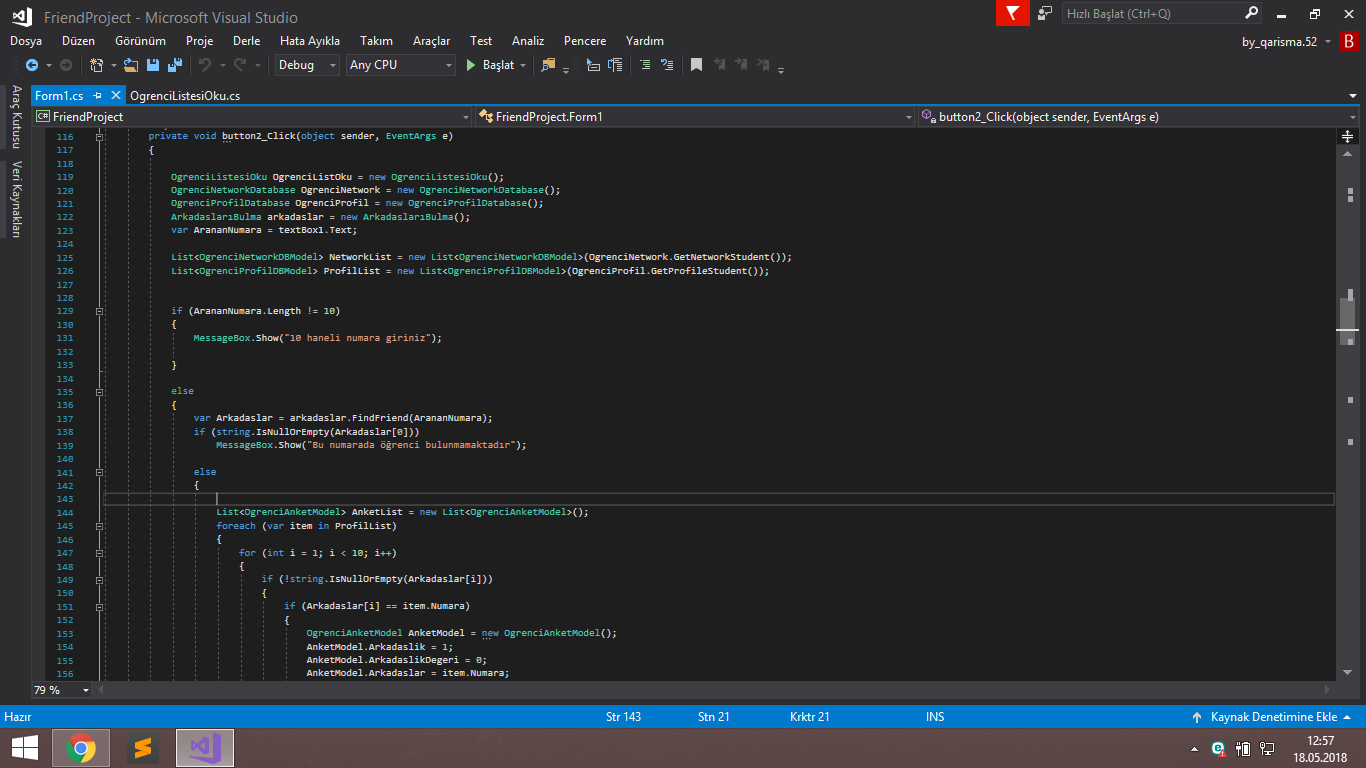
9-) OgrenciListesiOku Sınıfı : Bu sınıfta ogrenciListesi adlı öğrencilerin numaraları ve isimlerinin bulunduğu excel dosyayı okunup ilgili modelinin içine atılır.



9-) Deger Metodu : Bu metod anket model tipinde bir parametre alır ve gelen parametre değerine göre sadece profil anket bilgilerini pars ederek geriye bu değerlerii döndürür.

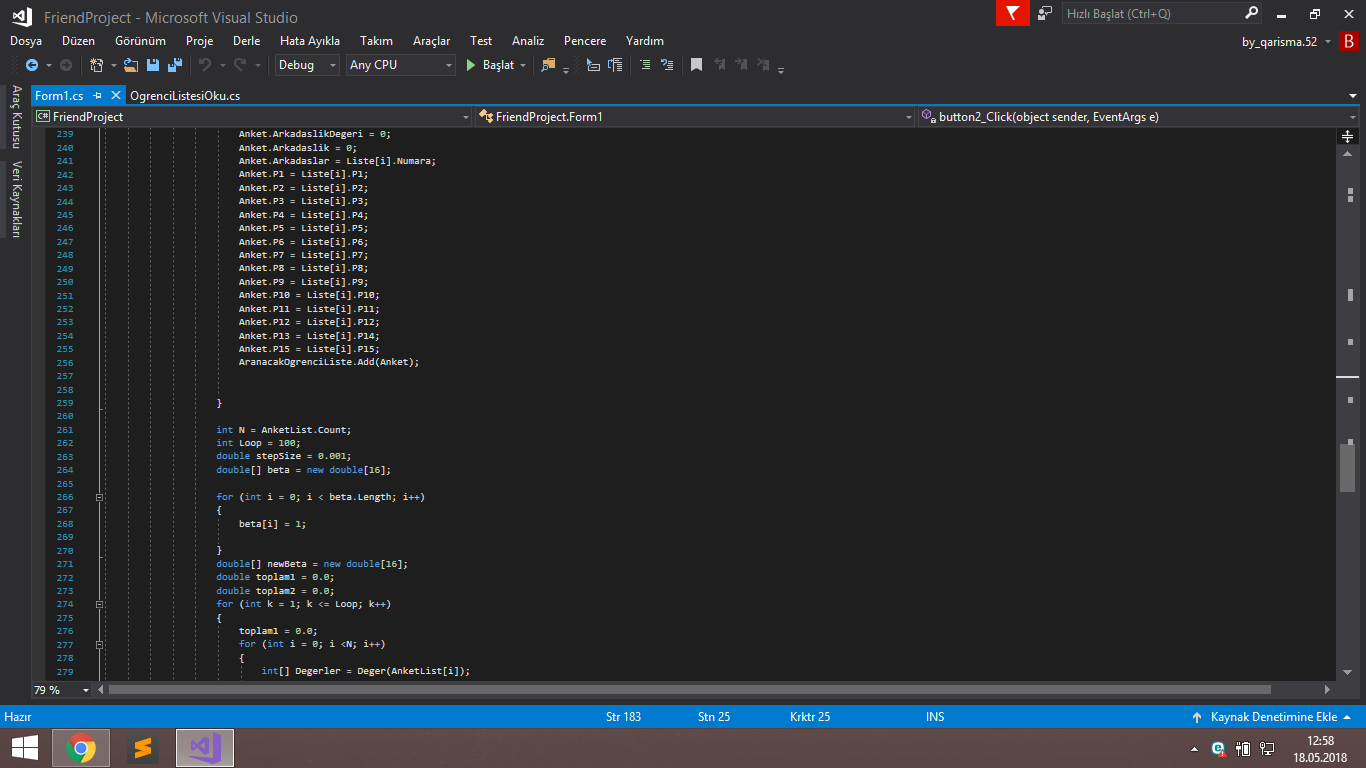


10-) Button2 Click Metodu : Bu butonun click olayında textboxa girilen numara alınır ve o numaraya göre lojitik regresyon yapılıp 10 tane arkadaş önerilir.Ayrıntılar kaynak kod açıklamalarında verilecektir.



## 

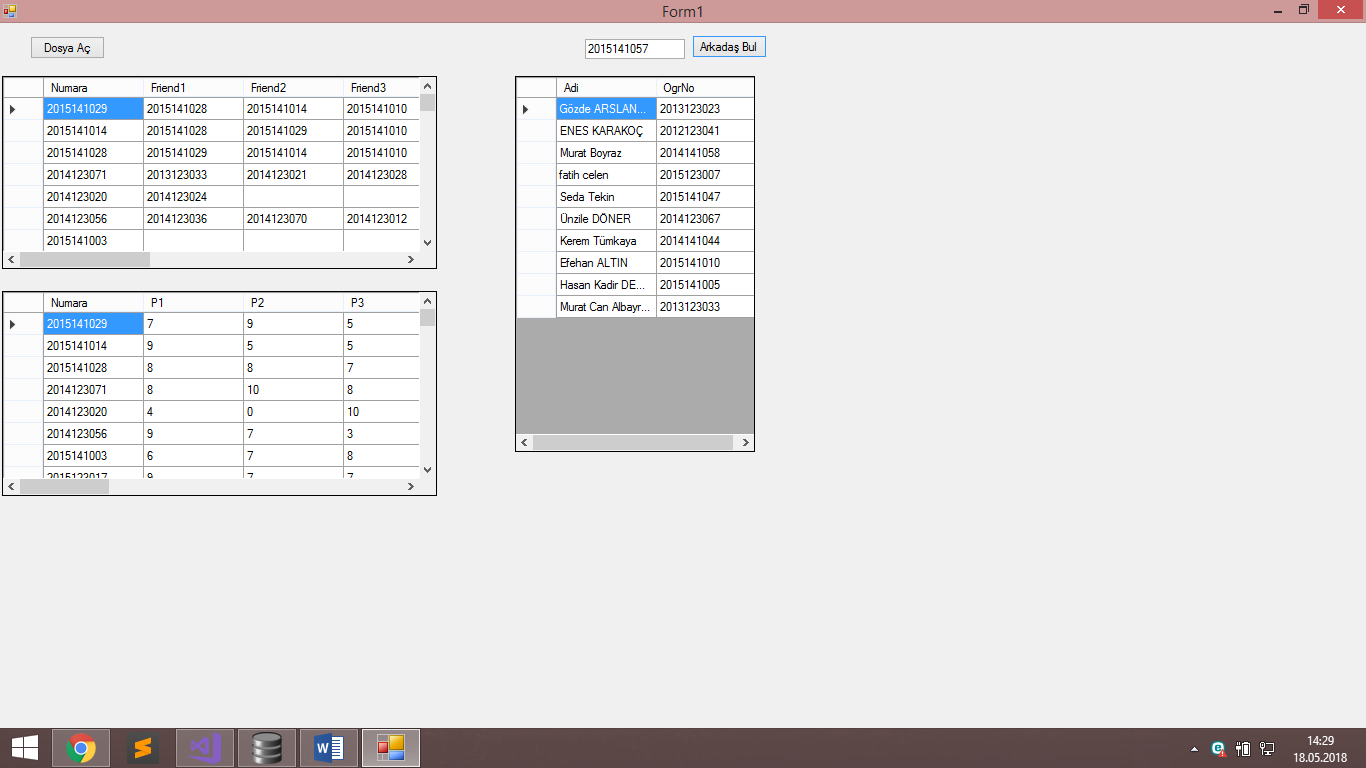
## 



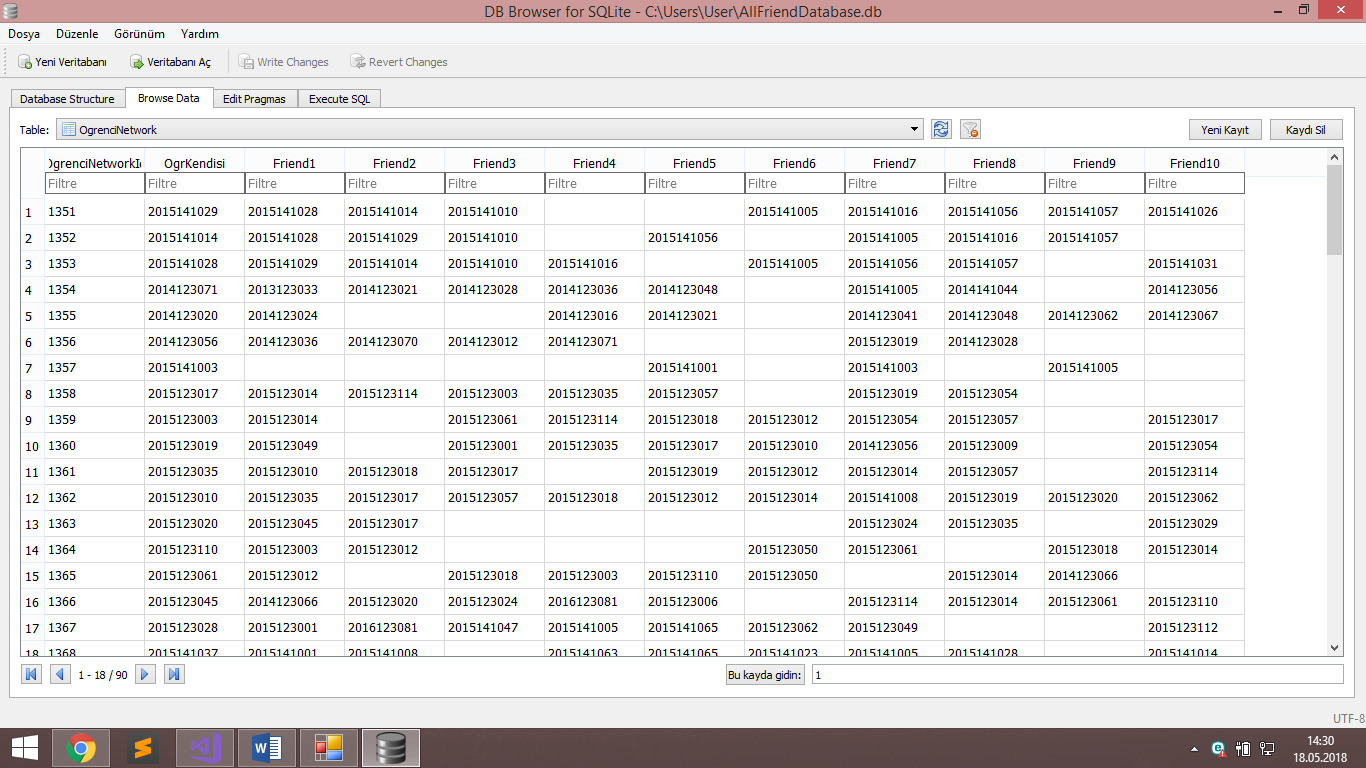
## 

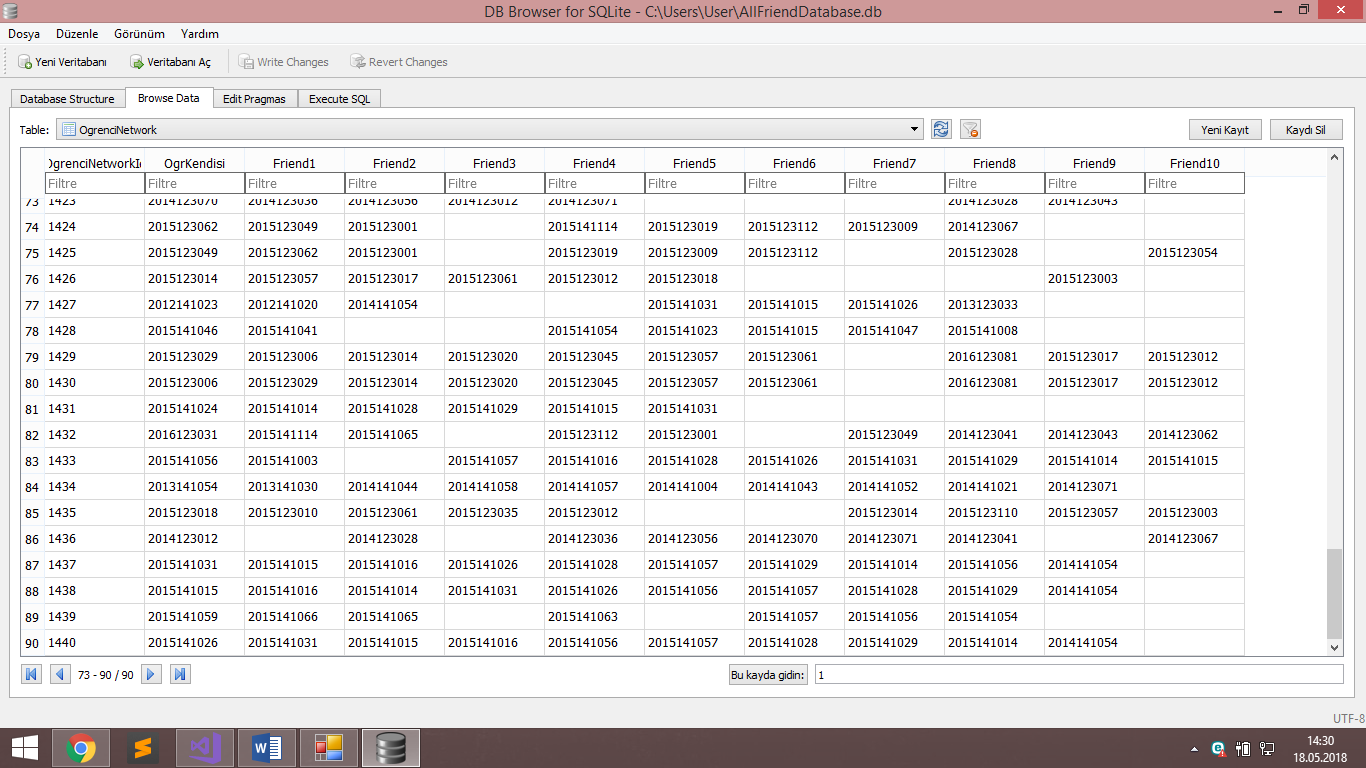
## 

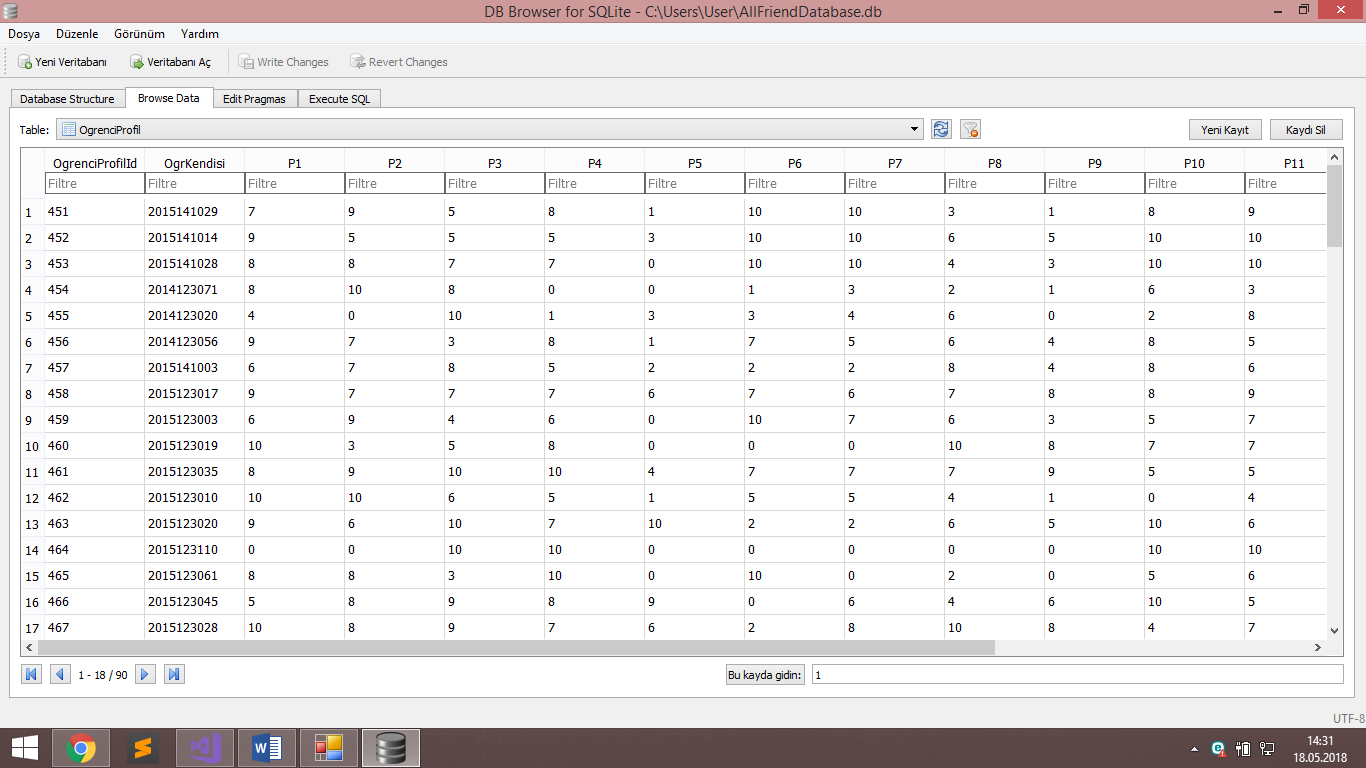
## SONUÇLAR

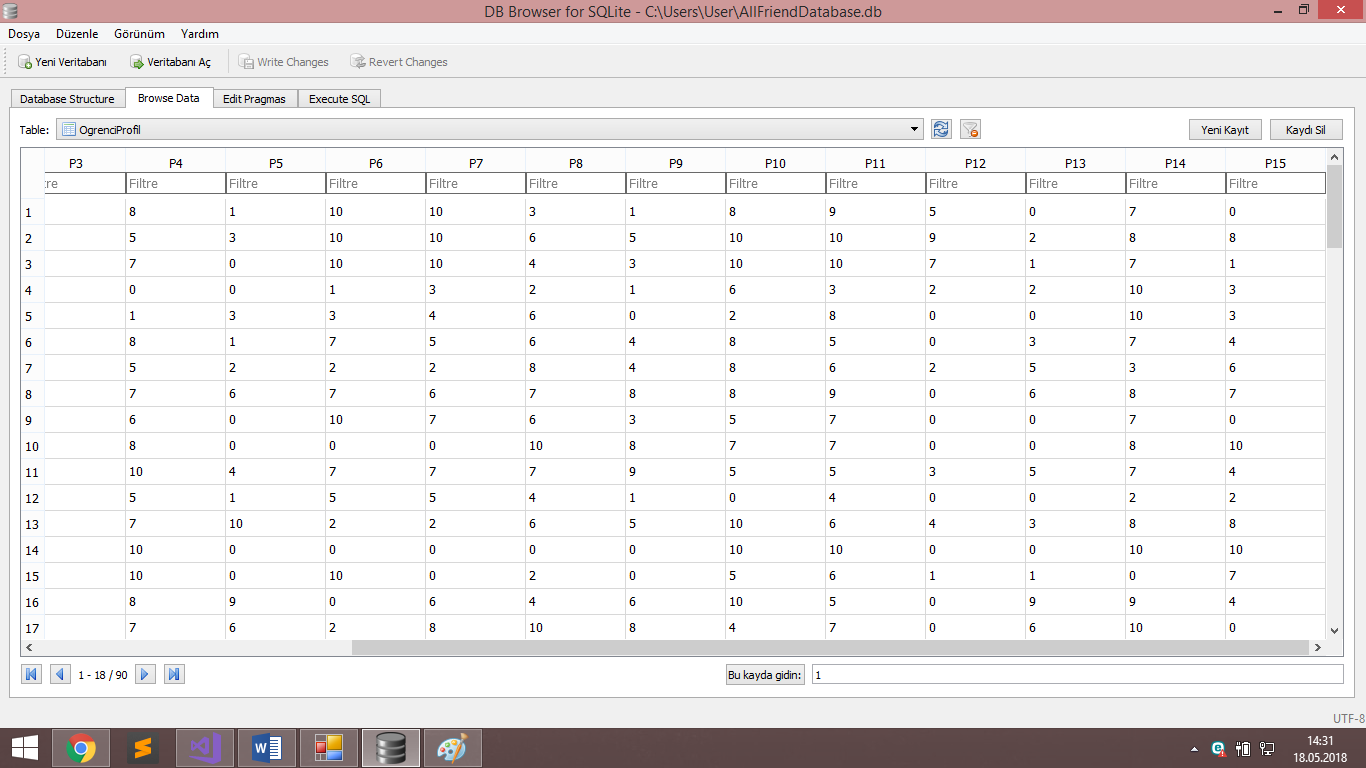


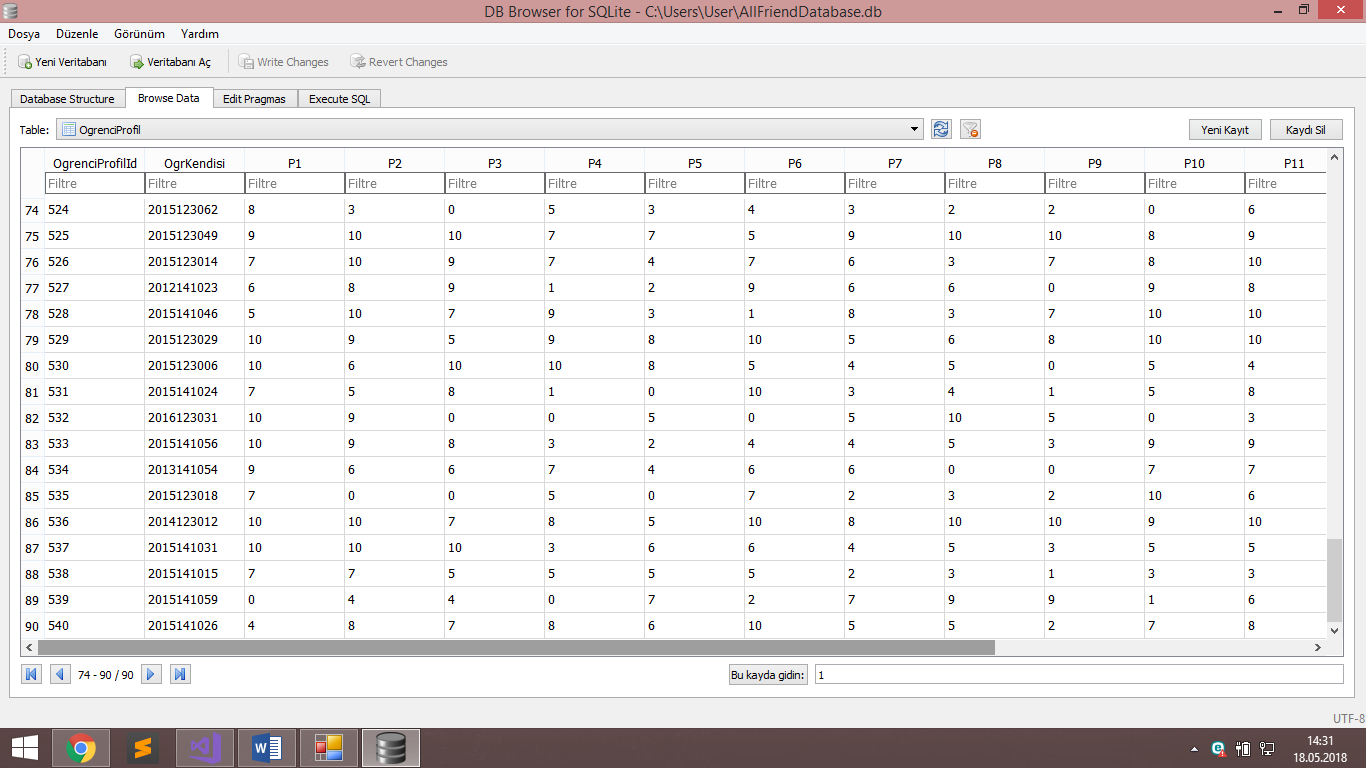
Database Çıktıları :











## KAYNAK KODLAR

using System;

using System.Collections.Generic;

using System.Data.SQLite;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FriendProject.Database

{

public class ArkadaslarıBulma

{

public string[] FindFriend(string number)//Bu fonksiyon textbox'a girilen numaraya göre OgrenciNetwork tablosunda arama yapar o numaranın arkadaşları bulur ve geri döndürür.

{

string path = @"C:\Users\User\AllFriendDatabase.db";

string[] Friends = new string[11];

SQLiteConnection con = new SQLiteConnection("DataSource =" + path);

con.Open();

SQLiteCommand cmd = new SQLiteCommand("select \* from OgrenciNetwork WHERE OgrKendisi = @no", con);

cmd.Parameters.AddWithValue("@no", number);

SQLiteDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

Friends[0] = dr.GetString(dr.GetOrdinal("OgrKendisi"));

Friends[1] = dr.GetString(dr.GetOrdinal("Friend1"));

Friends[2] = dr.GetString(dr.GetOrdinal("Friend2"));

Friends[3] = dr.GetString(dr.GetOrdinal("Friend3"));

Friends[4] = dr.GetString(dr.GetOrdinal("Friend4"));

Friends[5] = dr.GetString(dr.GetOrdinal("Friend5"));

Friends[6] = dr.GetString(dr.GetOrdinal("Friend6"));

Friends[7] = dr.GetString(dr.GetOrdinal("Friend7"));

Friends[8] = dr.GetString(dr.GetOrdinal("Friend8"));

Friends[9] = dr.GetString(dr.GetOrdinal("Friend9"));

Friends[10] = dr.GetString(dr.GetOrdinal("Friend10"));

}

con.Close();

return Friends;

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SQLite;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using FriendProject.DatabaseModel;

namespace FriendProject.Database

{

public class OgrenciNetworkDatabase

{

public void AddNetworkStudent(OgrenciNetworkDBModel ogrenci)//Button1 click olayında cağrılır ve click olayında okunan ogrencinetwork dosyasının içindeki veriler Veritabanındaki OgrenciNetwork tablosuna atılır.

{

string path = @"C:\Users\User\AllFriendDatabase.db";

SQLiteConnection con = new SQLiteConnection("DataSource =" + path);

con.Open();

SQLiteCommand cmd = new SQLiteCommand("insert into OgrenciNetwork(OgrKendisi,Friend1,Friend2,Friend3,Friend4,Friend5,Friend6,Friend7,Friend8,Friend9,Friend10) values(@no,@frd1,@frd2,@frd3,@frd4,@frd5,@frd6,@frd7,@frd8,@frd9,@frd10)", con);

cmd.Parameters.AddWithValue("@no", ogrenci.Numara);

cmd.Parameters.AddWithValue("@frd1", ogrenci.Friend1);

cmd.Parameters.AddWithValue("@frd2", ogrenci.Friend2);

cmd.Parameters.AddWithValue("@frd3", ogrenci.Friend3);

cmd.Parameters.AddWithValue("@frd4", ogrenci.Friend4);

cmd.Parameters.AddWithValue("@frd5", ogrenci.Friend5);

cmd.Parameters.AddWithValue("@frd6", ogrenci.Friend6);

cmd.Parameters.AddWithValue("@frd7", ogrenci.Friend7);

cmd.Parameters.AddWithValue("@frd8", ogrenci.Friend8);

cmd.Parameters.AddWithValue("@frd9", ogrenci.Friend9);

cmd.Parameters.AddWithValue("@frd10", ogrenci.Friend10);

cmd.ExecuteNonQuery();

con.Close();

}

public List<OgrenciNetworkDBModel> GetNetworkStudent()//Sonrasında gridview'e basmak amacıyla bu verilerin geri getirildiği fonksiyon.

{

List<OgrenciNetworkDBModel> Ogrenciler = new List<OgrenciNetworkDBModel>();

string path = @"C:\Users\User\AllFriendDatabase.db";

SQLiteConnection con = new SQLiteConnection("DataSource =" + path);

con.Open();

SQLiteCommand cmd = new SQLiteCommand("select \* from OgrenciNetwork", con);

SQLiteDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

OgrenciNetworkDBModel OgrenciNetworkModel = new OgrenciNetworkDBModel();

OgrenciNetworkModel.Numara = dr.GetString(dr.GetOrdinal("OgrKendisi"));

OgrenciNetworkModel.Friend1 = dr.GetString(dr.GetOrdinal("Friend1"));

OgrenciNetworkModel.Friend2= dr.GetString(dr.GetOrdinal("Friend2"));

OgrenciNetworkModel.Friend3 = dr.GetString(dr.GetOrdinal("Friend3"));

OgrenciNetworkModel.Friend4 = dr.GetString(dr.GetOrdinal("Friend4"));

OgrenciNetworkModel.Friend5 = dr.GetString(dr.GetOrdinal("Friend5"));

OgrenciNetworkModel.Friend6 = dr.GetString(dr.GetOrdinal("Friend6"));

OgrenciNetworkModel.Friend7 = dr.GetString(dr.GetOrdinal("Friend7"));

OgrenciNetworkModel.Friend8 = dr.GetString(dr.GetOrdinal("Friend8"));

OgrenciNetworkModel.Friend9 = dr.GetString(dr.GetOrdinal("Friend9"));

OgrenciNetworkModel.Friend10 = dr.GetString(dr.GetOrdinal("Friend10"));

Ogrenciler.Add(OgrenciNetworkModel);

}

con.Close();

return Ogrenciler;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using FriendProject.DatabaseModel;

using System.Data.SQLite;

namespace FriendProject.Database

{

public class OgrenciProfilDatabase//SQLite veritabanı kullanıldığında SQLite command kullanılmıştır.Projedeki bütün veritabanı işlemleri SQLite ile yapılır

{

public void AddProfileStudent(OgrenciProfilDBModel ogrenci)//Bu fonksiyonda ogrenciprofil tablosuna button1 click olayında açılan profil dosyasından gelen verilerin ekleme işlemleri yapılır

{

string path = @"C:\Users\User\AllFriendDatabase.db";

SQLiteConnection con = new SQLiteConnection("DataSource =" + path);

con.Open();

SQLiteCommand cmd = new SQLiteCommand("insert into OgrenciProfil(OgrKendisi,P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,P11,P12,P13,P14,P15) values(@no,@p1,@p2,@p3,@p4,@p5,@p6,@p7,@p8,@p9,@p10,@p11,@p12,@p13,@p14,@p15)", con);

cmd.Parameters.AddWithValue("@no", ogrenci.Numara);

cmd.Parameters.AddWithValue("@p1", ogrenci.P1);

cmd.Parameters.AddWithValue("@p2", ogrenci.P2);

cmd.Parameters.AddWithValue("@p3", ogrenci.P3);

cmd.Parameters.AddWithValue("@p4", ogrenci.P4);

cmd.Parameters.AddWithValue("@p5", ogrenci.P5);

cmd.Parameters.AddWithValue("@p6", ogrenci.P6);

cmd.Parameters.AddWithValue("@p7", ogrenci.P7);

cmd.Parameters.AddWithValue("@p8", ogrenci.P8);

cmd.Parameters.AddWithValue("@p9", ogrenci.P9);

cmd.Parameters.AddWithValue("@p10", ogrenci.P10);

cmd.Parameters.AddWithValue("@p11", ogrenci.P11);

cmd.Parameters.AddWithValue("@p12", ogrenci.P12);

cmd.Parameters.AddWithValue("@p13", ogrenci.P13);

cmd.Parameters.AddWithValue("@p14", ogrenci.P14);

cmd.Parameters.AddWithValue("@p15", ogrenci.P15);

cmd.ExecuteNonQuery();

con.Close();

}

public List<OgrenciProfilDBModel> GetProfileStudent() //Bu fonksiyonda ise eklenen veriler tablodan okunur(Gridviewe basmak için)

{

List<OgrenciProfilDBModel> Ogrenciler = new List<OgrenciProfilDBModel>();

string path = @"C:\Users\User\AllFriendDatabase.db";

SQLiteConnection con = new SQLiteConnection("DataSource =" + path);

con.Open();

SQLiteCommand cmd = new SQLiteCommand("select \* from OgrenciProfil", con);

SQLiteDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

OgrenciProfilDBModel OgrenciProfilModel = new OgrenciProfilDBModel();

OgrenciProfilModel.Numara = dr.GetString(dr.GetOrdinal("OgrKendisi"));

OgrenciProfilModel.P1 = dr.GetInt32(dr.GetOrdinal("P1"));

OgrenciProfilModel.P2 = dr.GetInt32(dr.GetOrdinal("P2"));

OgrenciProfilModel.P3 = dr.GetInt32(dr.GetOrdinal("P3"));

OgrenciProfilModel.P4 = dr.GetInt32(dr.GetOrdinal("P4"));

OgrenciProfilModel.P5 = dr.GetInt32(dr.GetOrdinal("P5"));

OgrenciProfilModel.P6 = dr.GetInt32(dr.GetOrdinal("P6"));

OgrenciProfilModel.P7 = dr.GetInt32(dr.GetOrdinal("P7"));

OgrenciProfilModel.P8 = dr.GetInt32(dr.GetOrdinal("P8"));

OgrenciProfilModel.P9 = dr.GetInt32(dr.GetOrdinal("P9"));

OgrenciProfilModel.P10 = dr.GetInt32(dr.GetOrdinal("P10"));

OgrenciProfilModel.P11 = dr.GetInt32(dr.GetOrdinal("P11"));

OgrenciProfilModel.P12 = dr.GetInt32(dr.GetOrdinal("P12"));

OgrenciProfilModel.P13 = dr.GetInt32(dr.GetOrdinal("P13"));

OgrenciProfilModel.P14 = dr.GetInt32(dr.GetOrdinal("P14"));

OgrenciProfilModel.P15 = dr.GetInt32(dr.GetOrdinal("P15"));

Ogrenciler.Add(OgrenciProfilModel);

}

con.Close();

return Ogrenciler;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FriendProject.DatabaseModel

{

public class OgrenciAnketModel//Regresyon için kullanılacak anket bilgilerinin ve arkadaslık değerlerinin tutulduğu model.

{

private string arkadas;

public string Arkadaslar

{

get { return arkadas; }

set { arkadas = value; }

}

private double arkadaslikDegeri;

public double ArkadaslikDegeri

{

get { return arkadaslikDegeri; }

set { arkadaslikDegeri = value; }

}

private int p1;

public int P1

{

get { return p1; }

set { p1 = value; }

}

private int p2;

public int P2

{

get { return p2; }

set { p2 = value; }

}

private int p3;

public int P3

{

get { return p3; }

set { p3 = value; }

}

private int p4;

public int P4

{

get { return p4; }

set { p4 = value; }

}

private int p5;

public int P5

{

get { return p5; }

set { p5 = value; }

}

private int p6;

public int P6

{

get { return p6; }

set { p6 = value; }

}

private int p7;

public int P7

{

get { return p7; }

set { p7 = value; }

}

private int p8;

public int P8

{

get { return p8; }

set { p8 = value; }

}

private int p9;

public int P9

{

get { return p9; }

set { p9 = value; }

}

private int p10;

public int P10

{

get { return p10; }

set { p10 = value; }

}

private int p11;

public int P11

{

get { return p11; }

set { p11 = value; }

}

private int p12;

public int P12

{

get { return p12; }

set { p12 = value; }

}

private int p13;

public int P13

{

get { return p13; }

set { p13 = value; }

}

private int p14;

public int P14

{

get { return p14; }

set { p14 = value; }

}

private int p15;

public int P15

{

get { return p15; }

set { p15 = value; }

}

private int arkadaslik;

public int Arkadaslik

{

get { return arkadaslik; }

set { arkadaslik = value; }

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FriendProject.DatabaseModel

{

public class OgrenciListesiModel//OgrenciListesi excel dosyanın değerlerinin tutulacağı model.

{

private string adi;

public string Adi {

get { return adi; }

set { adi = value; }

}

private string ogrNo;

public string OgrNo

{

get { return ogrNo; }

set { ogrNo = value; }

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FriendProject.DatabaseModel

{

/// <summary>

/// OgrenciNetwork veritabanı tablosunun modelidir

/// Tablonun içi bu model ile doldulur veya tablo içindeki değerler alınır

/// </summary>

public class OgrenciNetworkDBModel

{

private string ogrKendisi;

public string Numara

{

get { return ogrKendisi; }

set { ogrKendisi = value; }

}

private string friend1;

public string Friend1

{

get { return friend1; }

set { friend1 = value; }

}

private string friend2;

public string Friend2

{

get { return friend2; }

set { friend2 = value; }

}

private string friend3;

public string Friend3

{

get { return friend3; }

set { friend3 = value; }

}

private string friend4;

public string Friend4

{

get { return friend4; }

set { friend4 = value; }

}

private string friend5;

public string Friend5

{

get { return friend5; }

set { friend5 = value; }

}

private string friend6;

public string Friend6

{

get { return friend6; }

set { friend6 = value; }

}

private string friend7;

public string Friend7

{

get { return friend7; }

set { friend7 = value; }

}

private string friend8;

public string Friend8

{

get { return friend8; }

set { friend8 = value; }

}

private string friend9;

public string Friend9

{

get { return friend9; }

set { friend9 = value; }

}

private string friend10;

public string Friend10

{

get { return friend10; }

set { friend10 = value; }

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FriendProject.DatabaseModel

{

/// <summary>

/// OgrenciProfil veritabanı tablosunun modelidir

/// Tablonun içi bu model ile doldulur veya tablo içindeki değerler alınır

/// </summary>

public class OgrenciProfilDBModel

{

private string numara;

public string Numara

{

get { return numara; }

set { numara = value; }

}

private int p1;

public int P1

{

get { return p1; }

set { p1 = value; }

}

private int p2;

public int P2

{

get { return p2; }

set { p2 = value; }

}

private int p3;

public int P3

{

get { return p3; }

set { p3 = value; }

}

private int p4;

public int P4

{

get { return p4; }

set { p4 = value; }

}

private int p5;

public int P5

{

get { return p5; }

set { p5 = value; }

}

private int p6;

public int P6

{

get { return p6; }

set { p6 = value; }

}

private int p7;

public int P7

{

get { return p7; }

set { p7 = value; }

}

private int p8;

public int P8

{

get { return p8; }

set { p8 = value; }

}

private int p9;

public int P9

{

get { return p9; }

set { p9 = value; }

}

private int p10;

public int P10

{

get { return p10; }

set { p10 = value; }

}

private int p11;

public int P11

{

get { return p11; }

set { p11 = value; }

}

private int p12;

public int P12

{

get { return p12; }

set { p12 = value; }

}

private int p13;

public int P13

{

get { return p13; }

set { p13 = value; }

}

private int p14;

public int P14

{

get { return p14; }

set { p14 = value; }

}

private int p15;

public int P15

{

get { return p15; }

set { p15 = value; }

}

}

}

using FriendProject.DatabaseModel;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Microsoft.Office.Interop.Excel;

using Excel = Microsoft.Office.Interop.Excel;

namespace FriendProject

{

public class OgrenciListesiOku//ogrenciListesi excel dosyanın okunduğu sınıftır.

{

public List<OgrenciListesiModel> getStudenList()//Bu fonksiyon ogrenciListesi excelini okur ve ogrenciListesi modelinin içerisine atar.

{

List<OgrenciListesiModel> students = new List<OgrenciListesiModel>();

Excel.Application xlApp = new Excel.Application();

Excel.Workbook xlWorkbook = xlApp.Workbooks.Open(@"C:\Users\User\Desktop\ogrencilistesi.xlsx");

Excel.\_Worksheet xlWorksheet = (Excel.\_Worksheet)xlWorkbook.Sheets[1];

for (int i = 2; i <= xlWorksheet.UsedRange.Rows.Count; i++)

{

OgrenciListesiModel student = new OgrenciListesiModel();

student.OgrNo = (xlWorksheet.Cells[i, 1] as Range)?.Value2?.ToString();

student.Adi = (xlWorksheet.Cells[i, 2] as Range)?.Value2?.ToString();

students.Add(student);

}

xlWorkbook.Close();

return students;

}

}

}

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.IO;

using System.Windows.Forms;

using FriendProject.DatabaseModel;

using FriendProject.Database;

using System.Collections.ObjectModel;

namespace FriendProject

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)//Dosya aç butonu; bu butonda open file dialog ile kullanıcıya gerekli dosyalar açtırılır

{

OgrenciNetworkDatabase OgrenciNetwork = new OgrenciNetworkDatabase();//OgrenciProfil ve network dosyalarının içindeki verilerin veritabanı tablolarına eklenmesi veya geri istenmesi için

OgrenciProfilDatabase OgrenciProfil = new OgrenciProfilDatabase();//Gerekli nesneler oluşturuluyor.

OpenFileDialog openFile = new OpenFileDialog();//Dialog açtırılıyor...

openFile.Filter = @"CSV files (\*.csv)|\*.csv|XML files (\*.xml)|\*.xml";

openFile.DefaultExt = ".csv";

openFile.ShowDialog();

var SafeFileName = openFile.SafeFileName;

var ExcelFileName = openFile.FileName;

if (SafeFileName == "ogrenciNetwork.csv")//eğer dialogda seçilen dosyanın adı ogrenciNetwork ise

{

string[] NetworkRow = File.ReadAllLines(ExcelFileName);//Bu dosyanın bütün satırları okunuyor

for (int i = 0; i < NetworkRow.Length; i++)//satırların uzunluguna kadar dönecek for yani 90 kez.

{

OgrenciNetworkDBModel networkModel = new OgrenciNetworkDBModel();//Network modeli oluşturuluyor gelen veriler bu modelin içine atılıp ordan tabloya gidecek.

string[] AllStudents = NetworkRow[i].Split(',');//her bir satır virgüllere göre ayrılıp okunuyor.

networkModel.Numara = AllStudents[0];//satırın 0. elemanı yani ögrencinin kendisi geri kalan alanlar ise arkadasları bu işlem tüm satırlara uygulanıyor.

networkModel.Friend1 = AllStudents[1];

networkModel.Friend2 = AllStudents[2];

networkModel.Friend3 = AllStudents[3];

networkModel.Friend4 = AllStudents[4];

networkModel.Friend5 = AllStudents[5];

networkModel.Friend6 = AllStudents[6];

networkModel.Friend7 = AllStudents[7];

networkModel.Friend8 = AllStudents[8];

networkModel.Friend9 = AllStudents[9];

networkModel.Friend10 = AllStudents[10];

OgrenciNetwork.AddNetworkStudent(networkModel);//ardından dolan model network dosyalarını ekleyeceğimiz tabloya parametre olarak geçiliyor.

//yani OgrenciNetwork tablosu dosyadan okunan bütün verilerle düzenli bir şekilde doldu.

}

dataGridView1.DataSource = OgrenciNetwork.GetNetworkStudent();//ardından veritabanındaki kayıtlar grid view'e basılıyor.

}

if (SafeFileName == "ogrenciProfil.csv")//Ogrenci network için gerçekleşen tüm işlemlerin aynısı Ogrenci Profil içiinde gerçekleştiriliyor.

{

string[] ProfileRow = File.ReadAllLines(ExcelFileName);

for (int i = 0; i < ProfileRow.Length; i++)

{

OgrenciProfilDBModel ogrenciProfil = new OgrenciProfilDBModel();

string[] AllProfile = ProfileRow[i].Split(',');

ogrenciProfil.Numara = AllProfile[0];

ogrenciProfil.P1 = Convert.ToInt32(AllProfile[1]);

ogrenciProfil.P2 = Convert.ToInt32(AllProfile[2]);

ogrenciProfil.P3 = Convert.ToInt32(AllProfile[3]);

ogrenciProfil.P4 = Convert.ToInt32(AllProfile[4]);

ogrenciProfil.P5 = Convert.ToInt32(AllProfile[5]);

ogrenciProfil.P6 = Convert.ToInt32(AllProfile[6]);

ogrenciProfil.P7 = Convert.ToInt32(AllProfile[7]); ;

ogrenciProfil.P8 = Convert.ToInt32(AllProfile[8]);

ogrenciProfil.P9 = Convert.ToInt32(AllProfile[9]);

ogrenciProfil.P10 = Convert.ToInt32(AllProfile[10]);

ogrenciProfil.P11 = Convert.ToInt32(AllProfile[11]);

ogrenciProfil.P12 = Convert.ToInt32(AllProfile[12]);

ogrenciProfil.P13 = Convert.ToInt32(AllProfile[13]);

ogrenciProfil.P14 = Convert.ToInt32(AllProfile[14]);

ogrenciProfil.P15 = Convert.ToInt32(AllProfile[15]);

OgrenciProfil.AddProfileStudent(ogrenciProfil);

}

dataGridView2.DataSource = OgrenciProfil.GetProfileStudent();

}

}

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

{

}

private void button2\_Click(object sender, EventArgs e)//Bu button asıl işlemlerin gerçekleştiği buttondur.Arkadaş öneri buttonu

{

//Gerekli olan bütün sınıfların nesneleri üretiliyor

OgrenciListesiOku OgrenciListOku = new OgrenciListesiOku();

OgrenciNetworkDatabase OgrenciNetwork = new OgrenciNetworkDatabase();

OgrenciProfilDatabase OgrenciProfil = new OgrenciProfilDatabase();

ArkadaslarıBulma arkadaslar = new ArkadaslarıBulma();

var ArananNumara = textBox1.Text;//Textboxdaki numara değeri alınıyor.

List<OgrenciNetworkDBModel> NetworkList = new List<OgrenciNetworkDBModel>(OgrenciNetwork.GetNetworkStudent());//Network tablosundaki bütün veriler Network model tipinde bir listeye alınıyor

List<OgrenciProfilDBModel> ProfilList = new List<OgrenciProfilDBModel>(OgrenciProfil.GetProfileStudent());//Profil tablosundaki bütün veriler Profil model tipinde bir listeye alınıyor

if (ArananNumara.Length != 10)//eğer numara 10 haneden küçükse hata gösteriliyor.

{

MessageBox.Show("10 haneli numara giriniz");

}

else

{

var Arkadaslar = arkadaslar.FindFriend(ArananNumara);//eğer girilen numara hatasız ise oluşturduğumuş arkadaslar nesnesi yani veritabanında girilen numaraya göre arama yapan sınıfın nesnesi

//ve sınıfın içindeki arama fonksiyonu çağırılıyor textboxdaki arananNumara değeri parametre olarak gönderildi ve o numaraya göre arama yapıldı.

//Aranan numaranın arkadaşları da bulundu ve Arkadaslar adlı değişkene alındı.

if (string.IsNullOrEmpty(Arkadaslar[0]))//Eğer ilk değer yani numaranın kendisi boş geliyorsa böyle bir numara yok demektir bunun kontrolü sağlanıyor.

MessageBox.Show("Bu numarada öğrenci bulunmamaktadır");

else

{

List<OgrenciAnketModel> AnketList = new List<OgrenciAnketModel>();//Girilen numaranın arkadaş bilgilerinin alınacağı liste

foreach (var item in ProfilList)//Tüm profil bilgilerinin içerisinde sadece girilen numaranın arkadaşlarının profli bilgilerinin alınacağı döngü

{

for (int i = 1; i < 10; i++)//Girilen numaranın arkadaşlarının profil bilgilerinin alındığı döndü

{

if (!string.IsNullOrEmpty(Arkadaslar[i]))

{

if (Arkadaslar[i] == item.Numara)//Eğer arkadaslarının numarası ile profil listesindeki numaralar eşit ise

{

OgrenciAnketModel AnketModel = new OgrenciAnketModel();

AnketModel.Arkadaslik = 1;//Arkadaslık 1 olarak belirlendi çünkü arkadaşlar.

AnketModel.ArkadaslikDegeri = 0;

AnketModel.Arkadaslar = item.Numara;

AnketModel.P1 = item.P1;

AnketModel.P2 = item.P2;

AnketModel.P3 = item.P3;

AnketModel.P4 = item.P4;

AnketModel.P5 = item.P5;

AnketModel.P6 = item.P6;

AnketModel.P7 = item.P7;

AnketModel.P8 = item.P8;

AnketModel.P9 = item.P9;

AnketModel.P10 = item.P10;

AnketModel.P11 = item.P11;

AnketModel.P12 = item.P12;

AnketModel.P13 = item.P13;

AnketModel.P14 = item.P14;

AnketModel.P15 = item.P15;

AnketList.Add(AnketModel);//tüm arkadaslarının profil bilgilerini alır ve Anketlist'e atar

}

}

}

}

List<OgrenciProfilDBModel> Liste = new List<OgrenciProfilDBModel>();//ardından tekrar bütün profil bilgileri alınıyor

Liste = OgrenciProfil.GetProfileStudent();//Ve listenin içine atılıyor

OgrenciProfilDBModel ogrenciProfil = new OgrenciProfilDBModel();

for (int i = 0; i < Arkadaslar.Length; i++)//Numarası girilen kişinin arkadaşlarını bütün profil bilgilerinden çıkarır.

{

if (!string.IsNullOrEmpty(Arkadaslar[i]))

{

ogrenciProfil = Liste.FirstOrDefault(x => x.Numara == Arkadaslar[i]);

Liste.Remove(ogrenciProfil);

}

}

for (int i = 0; i < Liste.Count / 2; i++)//kalan profil bilglerinden ilk 40 kişi regresyon için alınır.

{

OgrenciAnketModel Anket = new OgrenciAnketModel();

Anket.ArkadaslikDegeri = 0;

Anket.Arkadaslik = 0;

Anket.Arkadaslar = Liste[i].Numara;

Anket.P1 = Liste[i].P1;

Anket.P2 = Liste[i].P2;

Anket.P3 = Liste[i].P3;

Anket.P4 = Liste[i].P4;

Anket.P5 = Liste[i].P5;

Anket.P6 = Liste[i].P6;

Anket.P7= Liste[i].P7;

Anket.P8 = Liste[i].P8;

Anket.P9 = Liste[i].P9;

Anket.P10 = Liste[i].P10;

Anket.P11 = Liste[i].P11;

Anket.P12 = Liste[i].P12;

Anket.P13 = Liste[i].P14;

Anket.P15 = Liste[i].P15;

AnketList.Add(Anket);

}

for (int i = 0; i < AnketList.Count; i++)//Regresyon için alınan 40 kişi de çıkarılır.

{

var AnketSil = Liste.FirstOrDefault(x => x.Numara == AnketList[i].Arkadaslar);

Liste.Remove(AnketSil);

}

List<OgrenciAnketModel> AranacakOgrenciListe = new List<OgrenciAnketModel>();

for (int i = 0; i < Liste.Count; i++)//Ardından hiç dokunulmamış 40 kişi karşılaştırma yapılmak için bir listeye alınır

{

OgrenciAnketModel Anket = new OgrenciAnketModel();

Anket.ArkadaslikDegeri = 0;

Anket.Arkadaslik = 0;

Anket.Arkadaslar = Liste[i].Numara;

Anket.P1 = Liste[i].P1;

Anket.P2 = Liste[i].P2;

Anket.P3 = Liste[i].P3;

Anket.P4 = Liste[i].P4;

Anket.P5 = Liste[i].P5;

Anket.P6 = Liste[i].P6;

Anket.P7 = Liste[i].P7;

Anket.P8 = Liste[i].P8;

Anket.P9 = Liste[i].P9;

Anket.P10 = Liste[i].P10;

Anket.P11 = Liste[i].P11;

Anket.P12 = Liste[i].P12;

Anket.P13 = Liste[i].P14;

Anket.P15 = Liste[i].P15;

AranacakOgrenciListe.Add(Anket);

}

int N = AnketList.Count;//Regresyon için alınan 40 kişi

int Loop = 100;//iterasyon sayısı

double stepSize = 0.001;

double[] beta = new double[16];//Betaların değerlerinin tutulduğu dizi

for (int i = 0; i < beta.Length; i++)

{

beta[i] = 1;

}

double[] newBeta = new double[16];

double toplam1 = 0.0;

double toplam2 = 0.0;

for (int k = 1; k <= Loop; k++)

{

toplam1 = 0.0;

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

{

int[] Degerler = Deger(AnketList[i]);//regresyon için alınan 40 kişinin profil bilgileri bu dizide tutuluyor.

double hx = Math.Exp(-(beta[0] + beta[1] \* Degerler[1] + beta[2] \* Degerler[2] + beta[3] \* Degerler[3] + beta[4] \* Degerler[4] + beta[5] \* Degerler[5] +

beta[6] \* Degerler[6] + beta[7] \* Degerler[7] + beta[8] \* Degerler[8] + beta[9] \* Degerler[9] + beta[10] \* Degerler[10] + beta[11] \* Degerler[11] + beta[12] \* Degerler[12] +

beta[13] \* Degerler[13] + beta[14] \* Degerler[14] + beta[15] \* Degerler[15]));//Ardından bu değerlere göre regresyon uygulanır

double hb = 1.0 / (1.0 + hx);

double y = AnketList[i].Arkadaslik;//Arkadas olup olmadıkları durumu

toplam1 += hb - y;

}

newBeta[0] = beta[0] - (stepSize \* toplam1 / N);//Beta 0'ın yeni değerlerini bu dizide tutuyoruz

for (int t= 0; t < 15; t++)

{

toplam2 = 0.0;

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

{

int[] Degerler = Deger(AnketList[i]);//tekrardan regresyon için alınan ilk 40 kişinin profil değerleri alınıyor

double hb = 1.0 / (1.0 + Math.Exp(-(beta[0] + beta[1] \* Degerler[1] + beta[2] \* Degerler[2] + beta[3] \* Degerler[3] + beta[4] \* Degerler[4] + beta[5] \* Degerler[5] +

beta[6] \* Degerler[6] + beta[7] \* Degerler[7] + beta[8] \* Degerler[8] + beta[9] \* Degerler[9] + beta[10] \* Degerler[10] + beta[11] \* Degerler[11] + beta[12] \* Degerler[12] +

beta[13] \* Degerler[13] + beta[14] \* Degerler[14] + beta[15] \* Degerler[15])));

double y = AnketList[i].Arkadaslik;

toplam2 += (hb - y) \* Degerler[t];

}

newBeta[t] = beta[t] - (stepSize \* toplam2 / N);//Beta0 hariç diğer tüm betaların yeni değerlerini burda tutuyoruz

}

for (int i = 0; i < beta.Length; i++)

{

beta[i] = newBeta[i];//Betalara yeni değerlerini atıyoruz

}

}

for (int i = 0; i < AranacakOgrenciListe.Count ; i++)//Ardından hiç dokunmadığımız 40 kişiye ait döngüye geliyoruz

{ //Regresyona aldığımız kişilerden çıkan beta değerlerine göre dokunmadığımız 40 kişinin de değerlerini buluyoruz

int[] Degerler = Deger(AranacakOgrenciListe[i]);

double hb = 1.0 / (1.0 + Math.Exp(-(beta[0] + beta[1] \* Degerler[1] + beta[2] \* Degerler[2] + beta[3] \* Degerler[3] + beta[4] \* Degerler[4] + beta[5] \* Degerler[5] +

beta[6] \* Degerler[6] + beta[7] \* Degerler[7] + beta[8] \* Degerler[8] + beta[9] \* Degerler[9] + beta[10] \* Degerler[10] + beta[11] \* Degerler[11] + beta[12] \* Degerler[12] +

beta[13] \* Degerler[13] + beta[14] \* Degerler[14] + beta[15] \* Degerler[15])));

AranacakOgrenciListe[i].ArkadaslikDegeri = hb;//Bu değerleri hiç dokunmadığımız 40 kişinin arkadaşlık değerlerine atıyoruz

}

List<OgrenciAnketModel> BulunanArkadaslar = (from x in AranacakOgrenciListe//Ardından arkadaslık değeri en yüksek olan kişiler bulunuyor.

orderby x.ArkadaslikDegeri descending

select x).ToList();

var OgrenciListesi = OgrenciListOku.getStudenList();//Önerilen arkadasların adlarını da görüntülemek için Ogrencilistesi excelini okudugumuz Fonksiyonu cagırıyoruz

List<OgrenciListesiModel> OnerilenArkadasListe = new List<OgrenciListesiModel>();//Ogrencilistesi modelimizin tipinde bir liste olusturuyoruz 10 arkadas bu listeye alınacak.

for (int i = 0; i < BulunanArkadaslar.Count; i++)//Bulunan tüm arkadaşların sayısı kadar bir for

{

for (int j = 0; j < OgrenciListesi.Count; j++)//Ardından excel dosyasındaki tüm ögrenciler kadar dönecek bir for

{

if (BulunanArkadaslar[i].Arkadaslar == OgrenciListesi[j].OgrNo)//Eğer bulunan arkadaşların numaraları ile ögrenci listesindeki ögrencilerin numaraları eşit ise

{

if (OnerilenArkadasListe.Count < 10)//o eşit olanların ilk 10 tanesini alıyor.

{

OnerilenArkadasListe.Add(OgrenciListesi[j]);//ve bu ilk 10 kişieye OnerilenArkadas listesine atıyor.

}

}

}

}

dataGridView3.DataSource = OnerilenArkadasListe;//Ardından önerilen arkadaşlar grid view'e basılıyor.

}

}

}

private int[] Deger(OgrenciAnketModel anketDegeri)//Bu fonksiyon AnketModelin içinden sadece profil bilgilerini çekmemizi sağlar.

{

int[] parseDeger = new int[16];

parseDeger[0] = 0;

parseDeger[1] = anketDegeri.P1;

parseDeger[2] = anketDegeri.P2;

parseDeger[3] = anketDegeri.P3;

parseDeger[4] = anketDegeri.P4;

parseDeger[5] = anketDegeri.P5;

parseDeger[6] = anketDegeri.P6;

parseDeger[7] = anketDegeri.P7;

parseDeger[8] = anketDegeri.P8;

parseDeger[9] = anketDegeri.P9;

parseDeger[10] = anketDegeri.P10;

parseDeger[11] = anketDegeri.P11;

parseDeger[12] = anketDegeri.P12;

parseDeger[13] = anketDegeri.P13;

parseDeger[14] = anketDegeri.P14;

parseDeger[15] = anketDegeri.P15;

return parseDeger;

}

}

}