

T.C

TEKİRDAĞ NAMIK KEMAL ÜNİVERSİTESİ

ÇORLU MÜHENDİSLİK FAKÜLTESİ



BİLGİSAYAR MÜHENDİSLİĞİ

STAJ PROJE ÇALIŞMASI

AKADEMİSYEN VERİTABANI HAZIRLANMASI

1160606001 AZAT ATEŞ

DANIŞMAN

Dr. Öğr. Üyesi Erkan ÖZHAN

TEKİRDAĞ-2020

İÇİNDEKİLER

GİRİŞ.....	1
KISALTMALAR.....	1
VERİTABANI YAPISI	2
TABLolarIN YAPISI.....	3
TABLolar ARASI İLİŞKİLER.....	5
PROJE MİMARİSİ.....	5
KODLARIN ANALİZİ VE VERİTABANINA EKLENMESİ	6
KODUN ÇALIŞMA SİSTEMİ	22
TEMEL BAZI SORGULAR.....	23
VERİTABANININ DIŞARI/İÇERİ AKTARILMASI & LİNKLER & DETAYLAR	24
KAYNAKÇA.....	25

GİRİŞ

Proje Konusu: <http://hocaismi.cv.nku.edu.tr> formundaki sayfalardan akademisyenlere ait bilgileri çekip bu bilgilerden her akademisyene ait elde edilen bilgileri kullanarak bir SQL veritabanı tasarlanması. Oluşturulacak veritabanını kullanarak hangi akademisyenin hangi dönem hangi dersleri verdiği, hangi bölümde kaç tane akademisyen olduğu, hangi akademisyenin hangi idari görevleri yerine getirdiği gibi sorgular kolay bir şekilde yapılabilmesi ve akademisyen sayfaları ile tutarlı bilgiler üretmelidir. Tasarlanacak sistem için Docker image oluşturulup bu bu image Docker Hub'a yüklenecek. Ayrıca Docker Hub'a yüklenen image kullanılarak tasarlanan sistemin nasıl ayağa kaldırılacağını açıklayan bir dökümantasyon hazırlanacak. Bu dökümantasyonu Github hesabında ya da kişisel web sayfasında yayınlanacak.

KISALTMALAR

VS = Visual Studio

SQL = Structured Query Language

DB = Database

SSMS = SQL Server Management Studio

PK = Primary Key

FK = Foreign Key

UK = Unique Key

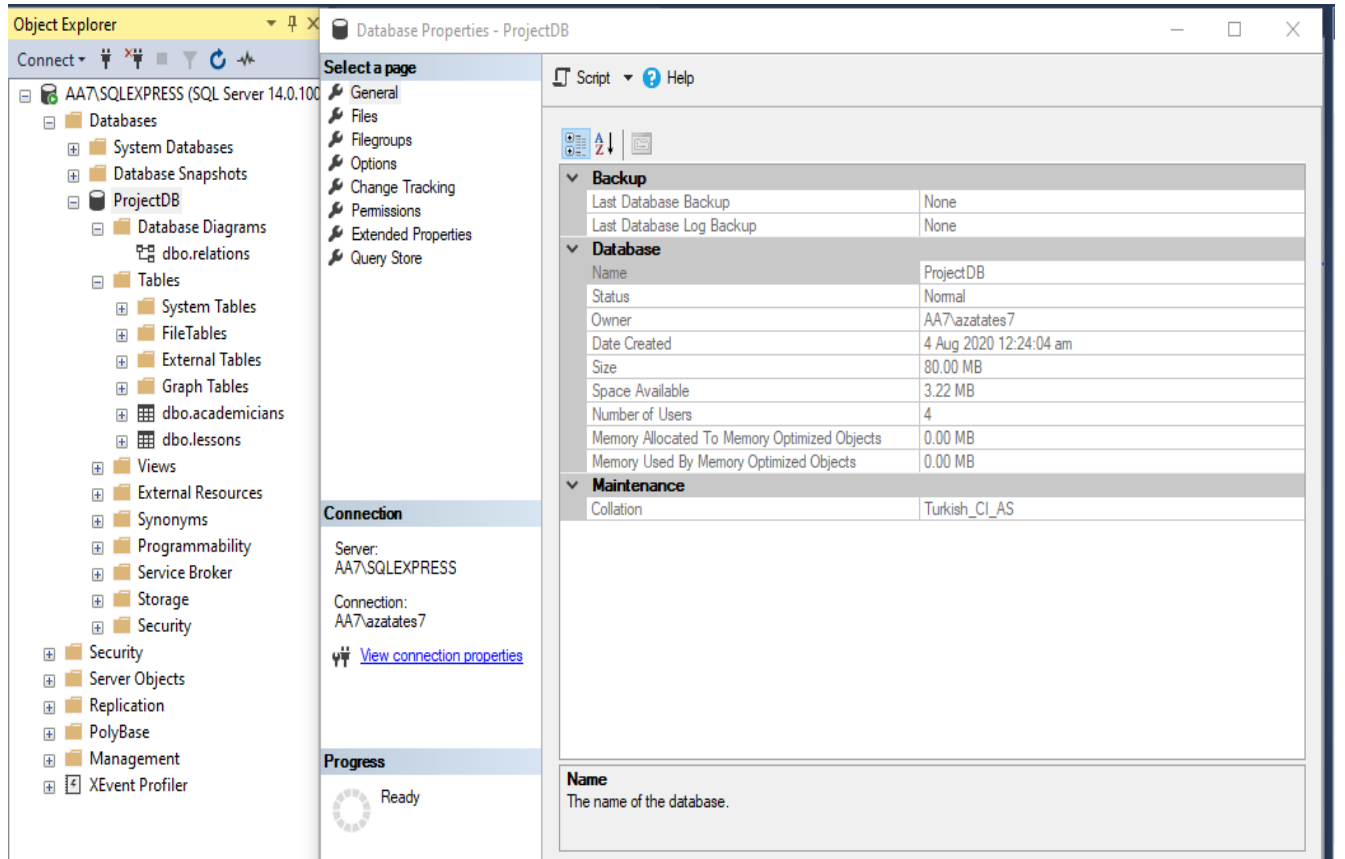
DF = Default

Dbo = DataBase Object

SP = Store Procedure

EXEC = Execute

VERİTABANI YAPISI



- Veritabanı **ProjectDB** adıyla oluşturuldu.
- Windows Authentication ile bağlantı sağlandı. (connectionString buna göre ayarlandı.)
- Karakter problem yaşanmaması için Collation Turkish_CI_AS seçildi.
- İki temel tablo oluşturuldu. (dbo.academicians, dbo.lessons)
- İlişkiler diyagramı relations adıyla oluşturuldu.
- Server Name = **AA7\SQLEXPRESS** username = **sa** password = **pRoJeCtPaSsWd**

TABLULARIN YAPISI

1. TABLO => dbo.academicians

Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
title	nvarchar(50)	<input type="checkbox"/>
name	nvarchar(50)	<input type="checkbox"/>
phone	nvarchar(50)	<input checked="" type="checkbox"/>
mail	nvarchar(50)	<input checked="" type="checkbox"/>
webaddress	nvarchar(50)	<input checked="" type="checkbox"/>
faculty	nvarchar(50)	<input checked="" type="checkbox"/>
branch	nvarchar(50)	<input checked="" type="checkbox"/>
main_branch	nvarchar(50)	<input checked="" type="checkbox"/>
university	nvarchar(50)	<input checked="" type="checkbox"/>
info_last_update_date	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Column Properties	
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes

[Tbl] dbo.academicians	
[Identity]	
(Name)	academicians
Database Name	ProjectDB
Description	
Schema	dbo
Server Name	aa7\sqlexpress
Table Designer	
Identity Column	id
Indexable	Yes
Lock Escalation	Table
Regular Data Space PRIMARY	
(Data Space Type)	Filegroup
Filegroup or Part	PRIMARY
Partition Column	
Replicated	No
Row GUID Column	
Text/Image Filegroup	PRIMARY

- Akademisyene ait temel bilgileri içeren tablo.
- Ayrıca son güncelleme bilgisini içeren alan eklendi.
- Id, INT ve otomatik artan yapıda.
- university alanı default olarak (N'Tekirdağ Namık Kemal Üniversitesi') seçili ve kodlamada bu alan eklenmiyor default değer direk tabloya ekleniyor.
- title, ünvanı içeren alan.
- branch, bölümü içeren alan.
- main_branch, ana bilim dalını alanını ifade ediyor.

2. TABLO => dbo.lessons

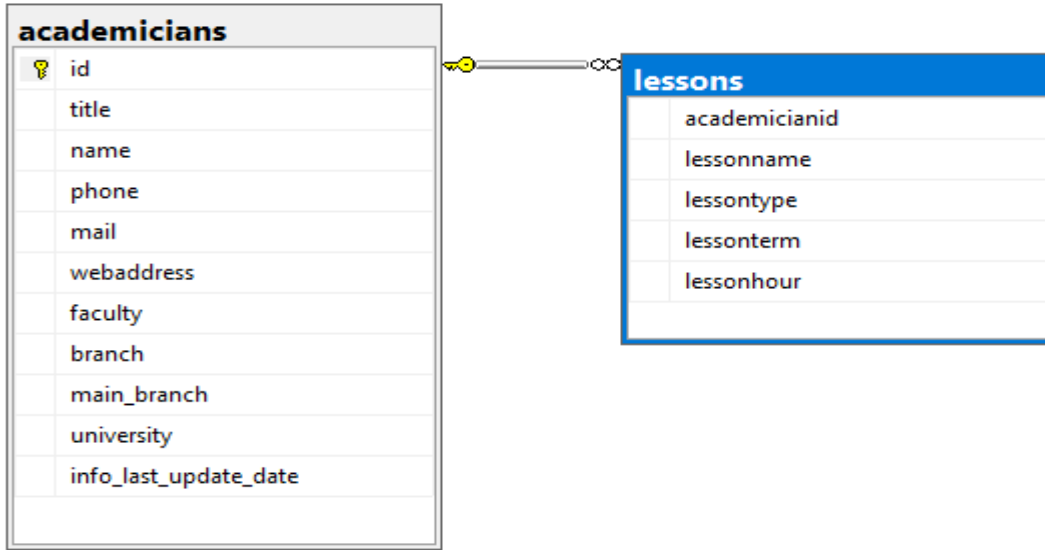
Column Name	Data Type	Allow Nulls
academicianid	int	<input type="checkbox"/>
lessonname	nvarchar(50)	<input checked="" type="checkbox"/>
lessonstype	nvarchar(50)	<input checked="" type="checkbox"/>
lessonterm	nvarchar(50)	<input checked="" type="checkbox"/>
lessonhour	nvarchar(2)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

[Tbl] dbo.lessons	
(Name)	lessons
Database Name	ProjectDB
Description	
Schema	dbo
Server Name	aa7\sqlexpress
Table Designer	
Identity Column	
Indexable	Yes
Lock Escalation	Table
Regular Data Space	PRIMARY
Replicated	No
Row GUID Column	
Text/Image Filegroup	PRIMARY

Column Properties	
(General)	
(Name)	academicianid
Allow Nulls	No
Data Type	int
Default Value or Binding	
Table Designer	

- Akademisyelere ait dersleri içeren tablo.
- academicianid alanı aynı anda çoklu insert içerdiği için primary key seçilmedi.
- lessonstype, dersin Doktora, Yüksek Lisans, Lisans ya da Ön Lisans olduğunu belirtir.
- Ders bilgisi yoksa kayıtlara sadece boş bir alan eklenir ve iki tablo arasındaki bağlantının devamı sağlanır.

TABLolar ARASI İlişKİLER



academics.id ile lessons.academicianid alanları ilişkilendirildi. (FK_lessons_academics)

PROJE MİMARİSİ

- **Programa Dili:** C#
- **Platform:** Console
- **Veritabanı:** MsSQL
- **VTYS:** MS SQL Server Management Studio
- **.NET Versiyon:** .Net Framework 4.6.1 (Net Core 3.1 ile Html Agility Pack uyumsuz)
- **Kullanılan Kütüphaneler & Teknolojiler**
 - Html Agility Pack (Web Scraping)**
 - Selenium (Web Scraping)**
 - Selenium Chrome Driver**
 - Regex (Karakter Eşleşmeleri)**
 - Dosya İşlemleri (Verileri Dosyaya Yazıp Verimli Olarak İşleme)**
 - Sql Connection (VS <-> SSMS Bağlantısı)**

KODLARIN ANALİZİ VE VERİTABANINA EKLENMESİ

(Bazı küçük alanlar kod bloklarında açıklama satırı olarak eklenmiştir.)

- Kod bloklarında # karakteri kullanılmamıştır.
- Temel kod içerisinde bazı alanlara // ile açıklama eklenmiştir.
- Aşağıda kodların yanında # sembolü ile kare içinde açıklama eklenmiştir.

```
using HtmlAgilityPack;
using System;
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using System.IO;
using System.Collections.ObjectModel;
using System.Collections.Generic;
using System.Text;
using System.Linq;
using System.Text.RegularExpressions;
using System.Data;
using System.Data.SqlClient;
```

Kullanılan temel kütüphaneler

```
namespace GetCvInfo
```

```
{
```

Namespace adı

```
public class GetCvInfoMain
```

```
{
```

Class adı

```
public static string cvfile = @"cv_list.txt";
```

İşlemlerin yürütüldüğü ana dosya adı

```
public static void Main(string[] args)
```

```
{
```

Main Alanı

```
FileExistChecker(cvfile);
```

Dosya yoksa yarat fonksiyonu içeriği aşağıda gösterilmiştir

```
GetAllCvLines(); // fill cv_list.txt
```

Fakülte sayfalarından email alanlarını dosya içerisine yazan Fonksiyon. İçeriği aşağıda gösterilmiştir. Fonksiyon içerisinde bu bilgileri daha düzenli bir halde her kayıt için hocaismi.cv.nku.edu.tr olarak satır satır dosyaya ekleyen CvLinesAdder adlı bir fonksiyon da içerir.


```

int i = 1;

String mainxpath = @"/html/body/div[2]";
string lessonxpath = @"/html/body/div[2]/div[2]/div[2]";
String line = "";

try
{
    System.IO.StreamReader file = new System.IO.StreamReader("cv_list.txt");
    while ((line = file.ReadLine()) != null)
    {
        String tempfile2 = line.Split('.')[0] + ".txt";          # Örnek dosya adı lhtecer.cv.nku.edu.tr
        FileExistChecker(tempfile2);
        GetMainInfos(@"http://" + line, mainxpath, @"temp.txt");
        # Akademisyen bilgilerini düzenleyip veritabanına ekleyen fonksiyon
        GetLessonFiles(@"http://" + line, lessonxpath, tempfile2, i); i++;
        # Akademisyen ders bilgilerini düzenleyip veritabanına ekleyen fonksiyon
    }

    file.Close();
}
catch (Exception ex)
{
    Console.WriteLine("Error Detected : " + ex.Message);
}
finally
{
    //File.Delete(cvfile);
    # C:\Users\${username}\source\repos\{project_name}\{project_name}\bin\Debug\cv_file.txt

    Console.ReadKey();
}
}

private static void FileExistChecker(string filename) # Dosya yoksa yarat fonksiyon
{
    if (File.Exists(filename))
    {
        File.Delete(filename);
    }
    FileInfo f = new FileInfo(filename);
}
}

```

```

private static void GetAllCvLines(string filename = "temp_file.txt")
{
    FileExistChecker(filename);

    CvLinesAdder(@"http://cmf.nku.edu.tr/PersonelListesi/0/s/9790/801",
@"html/body/div[3]/div/div[2]/div[1]/div", filename);
    CvLinesAdder(@"http://gsf.nku.edu.tr/personel_/0/s/563/13393",
@"html/body/div[3]/div/div[2]/div[1]/div", filename);
    CvLinesAdder(@"http://dis.nku.edu.tr/Personel/0/s/11682/16256",
@"html/body/div[3]/div/div[2]/div[1]/div", filename);
    CvLinesAdder(@"http://hukuk.nku.edu.tr/Personel/0/s/9799/13736",
@"html/body/div[3]/div/div[2]/div[2]/div", filename);
    CvLinesAdder(@"http://iibf-ik.web.nku.edu.tr/AkademikKadro/0/s/3313/3369",
@"html/body/div[3]/div/div[2]/div", filename);
    CvLinesAdder(@"http://ilahiyat.nku.edu.tr/PersonelListesi/0/s/9804/8207",
@"html/body/div[3]/div/div[2]/div[2]/div", filename);
    CvLinesAdder(@"http://tip.nku.edu.tr/AkademikPersonel/0/s/455/22316",
@"html/body/div[3]/div/div[2]/div", filename);
    CvLinesAdder(@"http://veteriner.nku.edu.tr/akademikpersonel/0/s/7164/9090",
@"html/body/div[3]/div/div[2]/div[2]", filename);
    CvLinesAdder(@"http://ziraat.nku.edu.tr/PersonelListesi/0/s/9860/684",
@"html/body/div[3]/div/div[2]/div[1]/div", filename);
    CvLinesAdder(@"http://fened.nku.edu.tr/AkademikPersonel/0/s/329/19683",
@"html/body/div[3]/div/div[2]/div", filename);
}

```

```

private static void CvLinesAdder(string link, string xpath, string filename)
{

```

```

    FileExistChecker(filename);

```

```

    HtmlWeb web = new HtmlWeb();

```

```

    web.OverrideEncoding = Encoding.UTF8;

```

```

    HtmlDocument doc = web.Load(link);

```

```

    List<HtmlNode> HeaderNames = doc.DocumentNode.SelectNodes(xpath).ToList();

```

```

    foreach (HtmlNode item in HeaderNames)
    {

```

```

        using (StreamWriter sw = File.AppendText(filename))
        {

```

```

            {

```

```

                sw.WriteLine(item.InnerText.Replace("&nbsp;", ""));
            }
        }
    }

```

```

    Regex regex = new Regex("([a-z]+)@([a-z]+)*");

```

```

    using (StreamReader reader = new StreamReader(filename))

```

Bazı sayfalarda

hocaismi.cv.nku.edu.tr alanı yok

email alanı @ sembolü üzerinden

parçalanarak hocaismi.cv.nku.edu.tr

haline getirilerek dosyaya eklenir.

```

{
    string line;
    while ((line = reader.ReadLine()) != null)
    {
        Match match = regex.Match(line);
        if (match.Success & line != "")
        {
            using (StreamWriter sw = File.AppendText(cvfile))
            {
                string[] parts = line.Trim().Split('@');
                string a = parts[0];
                string b = parts[1];
                sw.WriteLine((a + b.Replace(b, ".cv.nku.edu.tr")));
            }
        }
    }
}
File.Delete(filename);
}

private static void GetMainInfos(string url, string xpath, string filename)
{
    ChromeOptions chromeOptions = new ChromeOptions();
    chromeOptions.AddArguments(new List<string>() { "no-sandbox", "headless" });
    ChromeDriverService chromeDriverService = ChromeDriverService.CreateDefaultService();
    ChromeDriver driver = new ChromeDriver(chromeDriverService, chromeOptions);
    driver.Navigate().GoToUrl(url);

    ReadOnlyCollection<IWebElement> element = driver.FindElements(By.XPath(xpath));
    foreach (IWebElement item in element)
    {
        using (StreamWriter sw = File.AppendText(filename))
        {
            sw.WriteLine(item.Text);
        }
    }
    driver.Quit();
}

```

```

string title = "", name = "", phone = "", mail = url.Replace("http://", "").Split('.')[0]
+ "@nku.edu.tr", webaddress = url.Trim(), faculty = "", branch = "", main_branch = "",
info_last_update_date = "N/A";
    int lineCount = 0;    string line = string.Empty;
    using (StreamReader readerlines = File.OpenText(filename))
    {
        while ((line = readerlines.ReadLine()) != null)
        {
            if (!line.Equals(string.Empty))
            {
                lineCount++;
            }
        }
    }
}

```

Geçersiz sayfalar(nku ana sayfasına yönlendirilen sayfalar) Kalite ifadesi ile başladığı için bu sayfalar veritabanına invalid page bilgisi ile eklenir.

```

string[] lines = File.ReadAllLines(filename);
name = lines[0].Replace(" NKUAVES", "").Trim();
if (name != "Kalite") // for catch invalid pages
{
    if (name.StartsWith("Araş. Gör. "))
    {
        title = "Araş. Gör. ";
        name = name.Replace(title, "|").Split('|')[1];
    }
    else if (name.StartsWith("Arş. Gör. Dr. "))
    {
        title = "Arş. Gör. Dr. ";
        name = name.Replace(title, "|").Split('|')[1];
    }
    else if (name.StartsWith("Arş. Gör. "))
    {
        title = "Arş. Gör. ";
        name = name.Replace(title, "|").Split('|')[1];
    }
}

```

{ "no-sandbox", "headless" }

no-sandbox timeout hatası için, headless tarayıcı açılmadan işlem arka plan üzerinden hızlıca yürütmesi için eklendi.

lineCount satır kontrolü için eklendi.

```

else if (name.StartsWith("Doç. Dr. "))
{
    title = "Doç. Dr. ";
    name = name.Replace(title, "|").Split('|')[1];
}
else if (name.StartsWith("Doç. "))
{
    title = "Doç. ";
    name = name.Replace(title, "|").Split('|')[1];
}
else if (name.StartsWith("Dr. Öğr. Üyesi "))
{
    title = "Dr. Öğr. Üyesi ";
    name = name.Replace(title, "|").Split('|')[1];
}
else if (name.StartsWith("Dr. "))
{
    title = "Dr. ";
    name = name.Replace(title, "|").Split('|')[1];
}
else if (name.StartsWith("MİMAR "))
{
    title = "MİMAR ";
    name = name.Replace(title, "|").Split('|')[1];
}
else if (name.StartsWith("Öğr. Gör. Dr. "))
{
    title = "Öğr. Gör. Dr. ";
    name = name.Replace(title, "|").Split('|')[1];
}
else if (name.StartsWith("Öğr. Gör. "))
{
    title = "Öğr. Gör. ";
    name = name.Replace(title, "|").Split('|')[1];
}
else if (name.StartsWith("Öğretmen "))
{
    title = "Öğretmen ";
    name = name.Replace(title, "|").Split('|')[1];
}
else if (name.StartsWith("Prof. Dr. "))
{
    title = "Prof. Dr. ";

```

```

        name = name.Replace(title, "|").Split('|')[1];
    }
    else if (name.StartsWith("Prof. "))
    {
        title = "Prof. ";
        name = name.Replace(title, "|").Split('|')[1];
    }
    else if (name.StartsWith("Serbest "))
    {
        title = "Serbest ";
        name = name.Replace(title, "|").Split('|')[1];
    }

    else
    {
        title = "Undefined";
    }

    if (lineCount > 20) // for check full columns
    {
        if (lines[15] == "") # info_last_update_date invalid
        {
            phone = lines[20].Trim().Replace("(", "").Replace(")", "").Replace(" ", "").Split(':')[1];
            faculty = lines[24].Split(':')[1].Trim();
            branch = lines[25].Trim().Split(':')[1];
            main_branch = lines[26].Trim().Split(':')[1];
        }
        else # info_last_update_date # info_last_update_date valid
        {
            Console.WriteLine(lines[20] + " : " + lines[15]);
            phone = lines[21].Trim().Replace("(", "").Replace(")", "").Replace(" ", "").Split(':')[1];
            faculty = lines[25].Split(':')[1].Trim();
            branch = lines[26].Trim().Split(':')[1];
            main_branch = lines[27].Trim().Split(':')[1];
            info_last_update_date = lines[15].Trim().Replace(" ", "").Split(':')[1];
        }
    }
}

```

```

else if (lineCount > 16 && lineCount < 20) // for check info_last_update_date valid
{
    info_last_update_date = lines[15].Trim().Replace(" ", "").Split(':')[1];
}
else // for check null
{

}

AddDatabase(title, name, phone, mail, webaddress, faculty, branch, main_branch,
info_last_update_date);
}
else
{
    AddDatabase("Invalid", "Page", phone, mail, webaddress, faculty, branch, main_branch,
info_last_update_date);
}
File.Delete(filename);
driver.Quit();
}

private static void AddDatabase(string title, string name, string phone, string mail, string
webaddress, string faculty, string branch, string main_branch, string info_last_update_date)
{
    String connectionstring = @"Server=AA7\SQLEXPRESS; Database=ProjectDB;
Trusted_Connection=True;";
    string InsertQuery = @"INSERT INTO dbo.academicians(title, name, phone, mail, webaddress,
faculty, branch, main_branch, info_last_update_date) VALUES(@title, @name, @phone, @mail,
@webaddress, @faculty, @branch, @main_branch, @info_last_update_date)";

try
{
    using (SqlConnection connection = new SqlConnection(connectionstring))
    {

```

```

using (SqlCommand command = new SqlCommand(InsertQuery, connection))
{
    command.Parameters.Add("@title", SqlDbType.NChar).Value = title;
    command.Parameters.Add("@name", SqlDbType.NChar).Value = name;
    command.Parameters.Add("@phone", SqlDbType.NChar).Value = phone;
    command.Parameters.Add("@mail", SqlDbType.NChar).Value = mail;
    command.Parameters.Add("@webaddress", SqlDbType.NChar).Value = webaddress;
    command.Parameters.Add("@faculty", SqlDbType.NChar).Value = faculty;
    command.Parameters.Add("@branch", SqlDbType.NChar).Value = branch;
    command.Parameters.Add("@main_branch", SqlDbType.NChar).Value = main_branch;
    command.Parameters.Add("@info_last_update_date", SqlDbType.NChar).Value =
info_last_update_date;

    connection.Open();
    command.ExecuteNonQuery();
    connection.Close();
}
}
}
catch (Exception ex)
{
    Console.WriteLine("Error Detected : " + ex.Message);
}
}

private static void GetLessonFiles(string url, string xpath, string filename, int i)
{
    FileExistChecker(filename);

    ChromeOptions chromeOptions = new ChromeOptions();
    chromeOptions.AddArguments(new List<string>() { "no-sandbox", "headless" });
    ChromeDriverService chromeDriverService = ChromeDriverService.CreateDefaultService();
    ChromeDriver driver = new ChromeDriver(chromeDriverService, chromeOptions);
    driver.Navigate().GoToUrl(url);
    using (StreamWriter sw = File.AppendText(filename))
    {
        sw.WriteLine("empty line for create file");
    }
    try
    {
        string html = @"/html/body/div[2]/div[2]/div[1]/ul/li[6]/a";
        string outhtml = driver.FindElement(By.XPath(html)).GetAttribute("outerHTML");
        driver.FindElement(By.LinkText("Verdiği Dersler")).Click();
    }
}

```



```

        string funcname = outhterhtml.Substring(32, 40).Replace("#dersler\" onclick=\"",
        "").Split(';')[0].ToString();
        Boolean result = funcname.Any(char.IsDigit); // for check function return null
        if (result)
        {
            ((IJavaScriptExecutor)driver).ExecuteScript(funcname);
            ((ITakesScreenshot)driver).GetScreenshot().SaveAsFile(i + ".png"); // for trigger function
            ReadOnlyCollection<IWebElement> element =
driver.FindElements(By.XPath(xpath));
            foreach (IWebElement item in element)
            {
                using (StreamWriter sw = File.AppendText(filename))
                {
                    sw.WriteLine(item.Text);
                }
            }
            File.Delete(i + ".png");
            FileAnalyzer(filename, i);
        }
        else
        {
            AddDatabase2(i, "", "", "", "");
        }
    }
    catch (Exception)
    {
        AddDatabase2(i, "", "", "", "");
    }
    finally
    {
        driver.Quit();
        File.Delete(filename);
    }
}

```

#LinkText ile dersler alanı açılır ama veriler onclick içerisindeki fonksiyon kullanılır.

Sayfalar da onclick içerisinde DersGetir('1', '2') tarzı içeriğindeki rakamlar değişen bir Fonksiyon var.

outhterHTML ile değişen Fonksiyon elde edilir.

ITakesScreenshot ekran görüntüsü almaya yarayan bir kütüphane bu çalışmada fonksiyonu tetikleyip elementlerin kontrolünü sağlar. Bu alan çalıştırılmazsa fonksiyon pasif hale düşer.

```

private static void FileAnalyzer(string filename, int i)
{
    string lessonname = "", lesstontype = "", lessonterm = "", lessonhour = "";
    string[] lines = File.ReadAllLines(filename);
    int l1 = 0, l2 = 0, l3 = 0, l4 = 0;
    int lineslength = lines.Length;
    for (int k = 0; k < lineslength; k += 1)
    {
        string liness = lines[k].Trim();
    }
}

```

```

if (liness == "Doktora")
{
    l1 = k;
}
else if (liness == "Yüksek Lisans")
{
    l2 = k;
}
else if (liness == "Lisans")
{
    l3 = k;
}
else if (liness == "Ön Lisans")
{
    l4 = k;
}
else
{
}
}
}

```

```

int plus = l1 + l2 + l3 + l4;
if (plus == 0)
{
    AddDatabase2(i, lessonname, lesstertype, lessonterm, lessonhour);
}
if ((l1 > 0) & (l2 > 0) & (l3 > 0) & (l4 > 0))
{
    for (int lessonline = l1 + 2; lessonline < l2; lessonline++)
    {
        lesstertype = "Doktora";
        lessonterm = lines[lessonline].Split(' ')[0];
        lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
        lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
        AddDatabase2(i, lessonname, lesstertype, lessonterm, lessonhour);
    }
}

```

```

for (int lessonline = l2 + 2; lessonline < l3; lessonline++)
{
    lessonstype = "Yüksek Lisans";
    lessonterm = lines[lessonline].Split(' ')[0];
    lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
    lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
    AddDatabase2(i, lessonname, lessonstype, lessonterm, lessonhour);
}
for (int lessonline = l3 + 2; lessonline < l4; lessonline++)
{
    lessonstype = "Lisans";
    lessonterm = lines[lessonline].Split(' ')[0];
    lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
    lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
    AddDatabase2(i, lessonname, lessonstype, lessonterm, lessonhour);
}
for (int lessonline = l4 + 2; lessonline < lineslength; lessonline++)
{
    lessonstype = "Ön Lisans";
    lessonterm = lines[lessonline].Split(' ')[0];
    lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
    lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
    AddDatabase2(i, lessonname, lessonstype, lessonterm, lessonhour);
}
}

```

```

else if ((l1 > 0) & (l2 > 0) & (l3 > 0) & (l4 < 1))
{
    for (int lessonline = l1 + 2; lessonline < l2; lessonline++)
    {
        lessonstype = "Doktora";
        lessonterm = lines[lessonline].Split(' ')[0];
        lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
        lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
        AddDatabase2(i, lessonname, lessonstype, lessonterm, lessonhour);
    }
}

```

```

for (int lessonline = l2 + 2; lessonline < l3; lessonline++)
{
    lesstertype = "Yüksek Lisans";
    lessonterm = lines[lessonline].Split(' ')[0];
    lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
    lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
    AddDatabase2(i, lessonname, lesstertype, lessonterm, lessonhour);
}
for (int lessonline = l3 + 2; lessonline < lineslength; lessonline++)
{
    lesstertype = "Lisans";
    lessonterm = lines[lessonline].Split(' ')[0];
    lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
    lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
    AddDatabase2(i, lessonname, lesstertype, lessonterm, lessonhour);
}
}
else if ((l1 > 0) & (l2 > 0) & (l3 < 1) & (l4 < 1))
{
    for (int lessonline = l1 + 2; lessonline < l2; lessonline++)
    {
        lesstertype = "Doktora";
        lessonterm = lines[lessonline].Split(' ')[0];
        lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
        lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
        AddDatabase2(i, lessonname, lesstertype, lessonterm, lessonhour);
    }
    for (int lessonline = l2 + 2; lessonline < lineslength; lessonline++)
    {
        lesstertype = "Yüksek Lisans";
        lessonterm = lines[lessonline].Split(' ')[0];
        lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
        lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
        AddDatabase2(i, lessonname, lesstertype, lessonterm, lessonhour);
    }
}
}

```

```

else if ((l1 > 0) & (l2 < 1) & (l3 < 1) & (l4 < 1))
{
    for (int lessonline = l1 + 2; lessonline < lineslength; lessonline++)
    {
        lesstertype = "Doktora";
        lessonterm = lines[lessonline].Split(' ')[0];
        lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
        lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
        AddDatabase2(i, lessonname, lesstertype, lessonterm, lessonhour);
    }
}
else if ((l1 < 1) & (l2 > 0) & (l3 > 0) & (l4 > 0))
{
    for (int lessonline = l2 + 2; lessonline < l3; lessonline++)
    {
        lesstertype = "Yüksek Lisans";
        lessonterm = lines[lessonline].Split(' ')[0];
        lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
        lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
        AddDatabase2(i, lessonname, lesstertype, lessonterm, lessonhour);
    }
    for (int lessonline = l3 + 2; lessonline < l4; lessonline++)
    {
        lesstertype = "Lisans";
        lessonterm = lines[lessonline].Split(' ')[0];
        lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
        lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
        AddDatabase2(i, lessonname, lesstertype, lessonterm, lessonhour);
    }
    for (int lessonline = l4 + 2; lessonline < lineslength; lessonline++)
    {
        lesstertype = "Ön Lisans";
        lessonterm = lines[lessonline].Split(' ')[0];
        lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
        lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
        AddDatabase2(i, lessonname, lesstertype, lessonterm, lessonhour);
    }
}
}

```

```

else if ((l1 < 1) & (l2 > 0) & (l3 > 0) & (l4 < 1))
{
    for (int lessonline = l2 + 2; lessonline < l3; lessonline++)
    {
        lesstertype = "Yüksek Lisans";
        lessonterm = lines[lessonline].Split(' ')[0];
        lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
        lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
        AddDatabase2(i, lessonname, lesstertype, lessonterm, lessonhour);
    }
    for (int lessonline = l3 + 2; lessonline < lineslength; lessonline++)
    {
        lesstertype = "Lisans";
        lessonterm = lines[lessonline].Split(' ')[0];
        lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
        lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
        AddDatabase2(i, lessonname, lesstertype, lessonterm, lessonhour);
    }
}
else if ((l1 < 1) & (l2 > 0) & (l3 < 1) & (l4 < 1))
{
    for (int lessonline = l2 + 2; lessonline < lineslength; lessonline++)
    {
        lesstertype = "Yüksek Lisans";
        lessonterm = lines[lessonline].Split(' ')[0];
        lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
        lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
        AddDatabase2(i, lessonname, lesstertype, lessonterm, lessonhour);
    }
}
}

```

```

else if ((l1 < 1) & (l2 < 1) & (l3 > 0) & (l4 > 0))
{
    for (int lessonline = l3 + 2; lessonline < l4; lessonline++)
    {
        lesstypetype = "Lisans";
        lessonterm = lines[lessonline].Split(' ')[0];
        lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
        lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
        AddDatabase2(i, lessonname, lesstypetype, lessonterm, lessonhour);
    }
    for (int lessonline = l4 + 2; lessonline < lineslength; lessonline++)
    {
        lesstypetype = "Ön Lisans";
        lessonterm = lines[lessonline].Split(' ')[0];
        lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
        lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
        AddDatabase2(i, lessonname, lesstypetype, lessonterm, lessonhour);
    }
}
else if ((l1 < 1) & (l2 < 1) & (l3 > 0) & (l4 < 1))
{
    for (int lessonline = l3 + 2; lessonline < lineslength; lessonline++)
    {
        lesstypetype = "Lisans";
        lessonterm = lines[lessonline].Split(' ')[0];
        lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
        lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
        AddDatabase2(i, lessonname, lesstypetype, lessonterm, lessonhour);
    }
}
else if ((l1 < 1) & (l2 < 1) & (l3 < 1) & (l4 > 0))
{
    for (int lessonline = l4 + 2; lessonline < lineslength; lessonline++)
    {
        lesstypetype = "Ön Lisans";
        lessonterm = lines[lessonline].Split(' ')[0];
        lessonhour = lines[lessonline].Substring(lines[lessonline].Length - 1);
        lessonname = lines[lessonline].Replace(lessonterm, "").Replace(lessonhour, "");
        AddDatabase2(i, lessonname, lesstypetype, lessonterm, lessonhour);
    }
}
}
}

```

```

private static void AddDatabase2(int i, string lessonname, string lesstertype, string lessonterm,
string lessonhour)
{
    String connectionstring = @"Server=AA7\SQLEXPRESS; Database=ProjectDB;
Trusted_Connection=True;";
    string InsertQuery = @"INSERT INTO dbo.lessons(academicianid, lessonname, lesstertype,
lessonterm, lessonhour) VALUES(@academicianid, @lessonname, @lesstertype, @lessonterm,
@lessonhour)";

    using (SqlConnection connection = new SqlConnection(connectionstring))
    {
        using (SqlCommand command = new SqlCommand(InsertQuery, connection))
        {
            command.Parameters.Add("@academicianid", SqlDbType.Int).Value = i;
            command.Parameters.Add("@lessonname", SqlDbType.NChar).Value = lessonname;
            command.Parameters.Add("@lesstertype", SqlDbType.NChar).Value = lesstertype;
            command.Parameters.Add("@lessonterm", SqlDbType.NChar).Value = lessonterm;
            command.Parameters.Add("@lessonhour", SqlDbType.NChar).Value = lessonhour;

            connection.Open();
            command.ExecuteNonQuery();
            connection.Close();
        }
    }
}

```


KODUN ÇALIŞMA SİSTEMİ

- 1- Tüm fakülte sayfalarındaki akademisyenlere ait mail bilgileri bir dosyaya yazılır.
- 2- Bu alanlar @ ile split edilip hocaismi.cv.nku.edu.tr olarak satır satır cv_list dosyasına eklenir.
- 3- Eklenen satırlardan ana sayfa linkine erişilir.
- 4- Linkdeki sayfa içeriği alanları ayrılarak veritabanına eklenir.
- 5- Linklerdeki dersler alanına erişilir.
- 6- Dersler parçalanarak veritabanına eklenir.

TEMEL BAZI SORGULAR

- `SELECT name FROM sys.foreign_keys`
- `SELECT * FROM dbo.lessons l ORDER BY l.academicianid DESC`
- `SELECT * FROM dbo.academicians a ORDER BY a.id DESC`
- `SELECT DISTINCT a.name FROM dbo.academicians a
INNER JOIN dbo.lessons l
ON a.id = l.academicianid
ORDER BY a.id DESC`
- `DECLARE @sql NVARCHAR(MAX);
SET @sql = N'';
SELECT @sql = @sql + QUOTENAME(s.name) + N'.' + QUOTENAME(t.name) +
QUOTENAME(c.name) + ';' + CHAR(13)
FROM sys.objects AS c
INNER JOIN sys.tables AS t
ON c.parent_object_id = t.[object_id]
INNER JOIN sys.schemas AS s
ON t.[schema_id] = s.[schema_id]
WHERE c.[type] IN ('D', 'C', 'F', 'PK', 'UQ')
ORDER BY c.[type];
PRINT @sql;`
- `SELECT a.name, l.lessonname, l.lessonterm
FROM dbo.academicians a
INNER JOIN dbo.lessons l
ON a.id = l.academicianid
WHERE l.lessonname LIKE '%Bilgisayar%' AND a.faculty LIKE 'Çorlu Müh%'
ORDER BY a.id ASC`
- `CREATE VIEW dbo.academician AS
SELECT *
FROM dbo.academicians a WITH (NOLOCK)
INNER JOIN dbo.lessons l WITH (NOLOCK)
ON a.id = l.academicianid
SELECT * FROM dbo.academician`
- `CREATE PROCEDURE dbo.ex_academician AS
BEGIN
SELECT DISTINCT a.id, a.title, a.name, a.info_last_update_date
FROM dbo.academicians a WITH (NOLOCK)
INNER JOIN dbo.lessons l WITH (NOLOCK)
ON a.id = l.academicianid
ORDER BY a.id DESC
END
EXEC dbo.ex_academician`

```
[dbo].[academicians][DF_academicians_university]  
[dbo].[lessons][FK_lessons_academicians]  
[dbo].[academicians][PK_academicians]  
[dbo].[sysdiagrams][UK_principal_name]
```

VERİTABANININ DIŞARI/İÇERİ AKTARILMASI

```
DECLARE @BackupFile NVARCHAR(255);  
SET @BackupFile = 'C:\database_backup.bak';  
PRINT @BackupFile;  
BACKUP DATABASE [ProjectDB] TO DISK = @BackupFile;
```

Docker ve SQL Server arasında yaşanan problemden dolayı direk veritabanından yukarıdaki script aracılığı ile veritabanı export edilmiş drive linkinde ve zip içerisinde yer alacaktır.

LİNKLER & DETAYLAR

- <https://stackoverflow.com/questions/1126571/import-export-database-with-sql-server-server-management-studio> linki aracılığı ile dosya içeri aktarılabilir.
- <https://drive.google.com/drive/folders/1OugaGX9MnIFAGTd-ErvGFH4YoC4sfSvn?usp=sharing> linki aracılığı ile kodlar, dökümanlar, veritabanı yedeği gibi dosyalara erişilebilir.
- Kaynakça da linkler sonradan eklenmiştir. Bu nedenle bazı linkler eklenememiştir.
- Kod çalıştırmada sorun yaşarsanız Program.cs yi kopyalarak yeni .NetFramework Console açarak Html Agility Pack, Selenium, Selenium Chrome Driverı Package Manager veya Nuget aracılığı ile yükleyerek projeyi sorunsuz çalıştırabilirsiniz.
- Veritabanının import edilmesinde bir sıkıntı yaşarsanız mail ile ulaşabilirsiniz.
- Drive üzerinden görüntüleme yetkisi genel olarak verilmiştir. Sorun yaşarsanız mail ile ulaşabilirsiniz.
- Kodlar sadece drive üzerinde diğer tüm ekler zip içerisinde yer alacaktır.
- Tüm dosyalar(zip dahil) drive üzerinde yedeklenmiştir.

KAYNAKÇA (LİNKLER SONRADAN EKLENMİŞTİR)

<https://stackoverflow.com/questions/19956533/sql-insert-query-using-c-sharp/19956944>

<https://sqlservergitimleri.com/sql-serverda-cannot-truncate-table-because-it-is-being-referenced-by-a-foreign-key-constraint-hatasi>

<https://www.broculos.net/2014/05/web-scraping-with-html-agility-pack.html>

<https://www.youtube.com/watch?v=LcSHky9b0AM>

<https://www.youtube.com/watch?v=TsQn3yAAgbE>

https://www.youtube.com/watch?v=nLadf49S_i8

<https://docs.microsoft.com/en-us/virtualization/windowscontainers/quick-start/set-up-environment?tabs=Windows-10-Client>

<https://forum.huawei.com/enterprise/en/getting-the-error-the-request-failed-or-the-service-did-not-respond-in-a-timely-fashion/thread/499563-885>

<https://stackoverflow.com/questions/1617402/the-request-failed-or-the-service-did-not-respond-in-a-timely-fashion>

<https://stackoverflow.com/questions/1126571/import-export-database-with-sql-server-server-management-studio>

<https://towardsdatascience.com/build-a-mssql-docker-container-800166ecca21>

<https://www.sqlshack.com/creating-your-own-sql-server-docker-image/>

<https://www.openmakesoftware.com/docker-image-multiple-repositories/>

<https://kohera.be/blog/sql-server/make-your-own-custom-sql-server-docker-container/>

<https://octopus.com/blog/running-sql-server-developer-install-with-docker>

<https://esha.zendesk.com/hc/en-us/articles/204947726-The-request-failed-or-the-service-did-not-respond-in-a-timely-fashion-Consult-the-event-log-or-other-applicable-error-logs-for-details-error-when-starting-SQL-Server-instance>

<https://docs.microsoft.com/en-us/sql/integration-services/import-export-data/start-the-sql-server-import-and-export-wizard?view=sql-server-ver15>

<https://www.youtube.com/watch?v=MUVCE550yw8>

<https://gokhansengun.com/docker-nedir-nasil-calisir-nerede-kullanilir/>

<https://buraksenyurt.com/post/Docker-SQL-Server-Kullan%C4%B1m%C4%B1>

<http://www.techbrothersit.com/2016/04/c-how-to-use-variables-into-sqlcommand.html>

<https://stackoverflow.com/questions/17716061/how-to-perform-control-save-operation-with-selenium-webdriver>

<https://dba.stackexchange.com/questions/90033/how-do-i-drop-all-constraints-from-all-tables>

<https://stackoverflow.com/questions/31057/how-to-insert-a-line-break-in-a-sql-server-varchar-nvarchar-string>

<https://stackoverflow.com/questions/41610860/how-to-use-onclick-event-in-selenium-webdriver>

<https://stackoverflow.com/questions/6229769/execute-javascript-using-selenium-webdriver-in-c-sharp>

<https://stackoverflow.com/questions/27267658/no-process-is-on-the-other-end-of-the-pipe-sql-server-2012/27271530>

<https://stackoverflow.com/questions/8075784/error-message-provider-shared-memory-provider-error-0-no-process-is-on-th>