

Anna Dubé

00091750

adube@my.athens.edu

Assignment 7

In this assignment, I created two applications. In both applications, I set up a database and read in data from a file to insert into the database. Then I built an interface that allows users to query the database. I took the user input to build a LINQ query, query the database, and display the results. The first application was implemented using WPF. The second application was implemented using an ASP.NET web application.

Problem 1:

```
<Window x:Class="DubeAssign7.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:DubeAssign7"
        mc:Ignorable="d"
        Title="MainWindow" Height="450" Width="800">
    <Canvas>
        <Label Canvas.Top="20" Canvas.Left="20" Content="Query"/>
        <TextBox Name="searchWhere" Canvas.Top="50" Canvas.Left="20" Height="30"
Width="175"/>
        <Button Name="search_button" Content="Search"
            Canvas.Top="100" Canvas.Left="20" Click="search_button_Click"/>
        <DataGrid Name="datag" Canvas.Top="20" Canvas.Left="250"/>
    </Canvas>
</Window>
```

```
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Data.SqlClient;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
```

[illegible]

```

        tempID += line[ss];
    }
    id.Add(tempID);

    for (x = ss; x < line.Length; x++)
    {
        if (line[x] == ' ' && x < line.Length - 1)
        {
            // get name
            if (char.IsLetter(line[x + 1]))
            {
                if (first) { first = false; }
                else first = true;

                tempN = "";
                for (int z = x + 1; z < line.Length; z++)
                {
                    tempN += line[z];
                    if (line[z] == ' ') { break; }
                }
                if (first) { fn.Add(tempN); }
                else if (!first) { ln.Add(tempN); }
            }
        }
        // batting average
        if (char.IsDigit(line[x]) && foundIt == false)
        {
            foundIt = true;
            tempBA = "";
            for (int z = x; z < line.Length; z++)
            {
                tempBA += line[z];
            }
            ba.Add(Convert.ToDouble(tempBA));
        }
    }
}

string insertQ = "";
//insert into database
for (int x = 0; x < id.Count; x++)
{
    insertQ += "INSERT INTO
Player(PlayerID,FirstName,LastName,BattingAverage) " +
"VALUES ('" + id[x] + "','" + fn[x] + "','" + ln[x] + "','" + ba[x] + "')";
}
com.CommandText = insertQ;
com.ExecuteNonQuery();
}
scon.Close();
}
}

private void search_button_Click(object sender, RoutedEventArgs e)
{
    found = true;
    string searchQ = searchWhere.Text;

```

```

string s1 = "", s2 = "";
bool switchString = false;
double batAv = 0.0;

for (int xx = 0; xx < searchQ.Length; xx++)
{
    if (searchQ[xx] == '=')
    {
        switchString = true;
    }
    if (searchQ[xx] == '=' || searchQ[xx] == ' ') { }
    else
    {
        if (!switchString) { s1 += searchQ[xx]; }
        else
        {
            s2 += searchQ[xx];
        }
    }
}

// set query
var query = from p in pe.Players
            where 1 == 2
            select p;
if (s1 == "BattingAverage")
{
    batAv = Convert.ToDouble(s2);
    query = from p in pe.Players
            where p.BattingAverage == batAv
            select p;
}
else if (s1 == "PlayerID")
{
    query = from p in pe.Players
            where p.PlayerID == s2
            select p;
}
else if (s1 == "FirstName")
{
    query = from p in pe.Players
            where p.FirstName == s2
            select p;
}
else if (s1 == "LastName")
{
    query = from p in pe.Players
            where p.LastName == s2
            select p;
}
if (query.Count() == 0)
{
    found = false;
}

// set dataGrid ItemsSource
DataTable dt = new DataTable("Player1");
dt.Columns.AddRange(new DataColumn[4] { new DataColumn("Player Id",
typeof(string)),

```

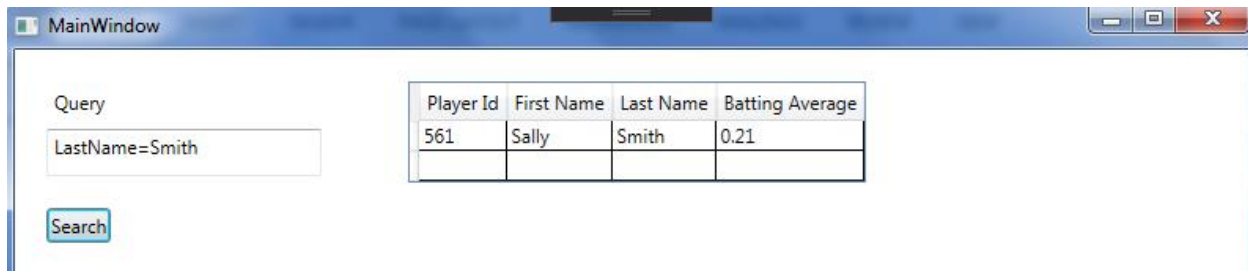
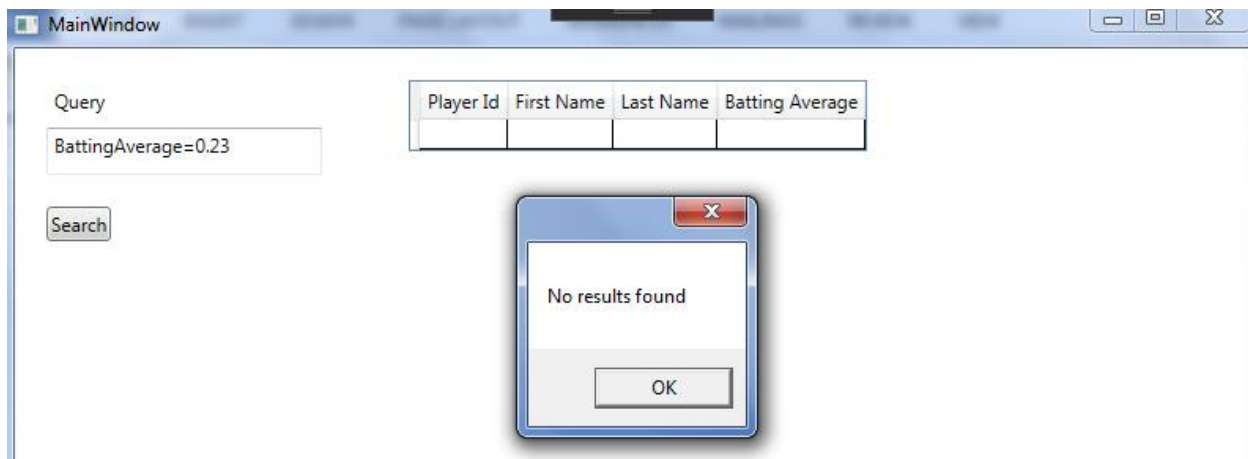
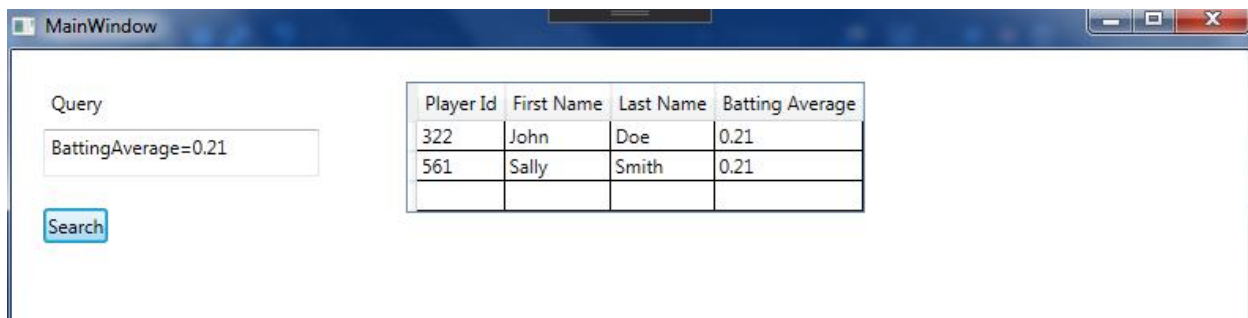
```

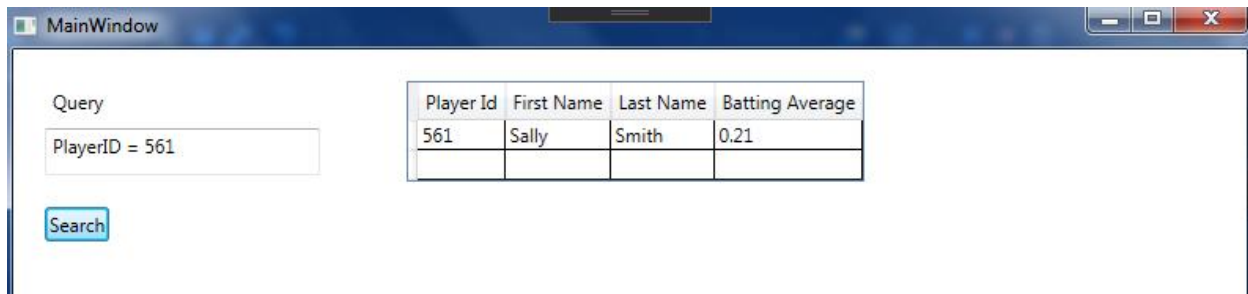
        new DataColumn("First Name", typeof(string)),
        new DataColumn("Last Name", typeof(string)),
        new DataColumn("Batting Average", typeof(float)) });

    foreach (Player p in query)
    {
        dt.Rows.Add(p.PlayerID, p.FirstName, p.LastName, p.BattingAverage);
    }
    datag.ItemsSource = dt.DefaultView;

    if (!found)
    {
        MessageBox.Show("No results found");
    }
}
}
}

```





Problem 2:

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="ITE365Assign7_2._Default"
%>
```

```
<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
    <div id="searchDiv" style="float: left; width: 300px; height: 300px;">
        <br />
        <label>Query</label>
        <input type="text" id="query" name="query" />
        <asp:Button runat="server" id="search" Text="Search" OnClick="search_Click"/>
    </div>

    <div id="tableDiv" style="float: left; width: 600px; height: 600px;">
        <br />
        <asp:Literal id="table1" runat="server"/>
    </div>
</asp:Content>
```

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.IO;
using System.Linq;
using System.Threading.Tasks;
using System.Text;
using System.Web.UI;

namespace ITE365Assign7_2
{
    public partial class _Default : Page
    {
        bool found = false;
        string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=
C:\Users\user\Documents\Players2.mdf;Integrated Security=True;Connect
Timeout=30";
        List<string> id = new List<string>();
        List<string> fn = new List<string>();
        List<string> ln = new List<string>();
        List<Double> ba = new List<Double>();
        protected void Page_Load(object sender, EventArgs e)
        {
```

```

using (SqlConnection scon = new SqlConnection(connectString))
{
    string createQuery = @"CREATE TABLE Player
        (PlayerID VARCHAR(50) NOT NULL PRIMARY KEY,
        FirstName VARCHAR(50) NULL,
        LastName VARCHAR(50) NULL,
        BattingAverage FLOAT NULL)";
    string dropQuery = @"DROP TABLE IF EXISTS Player";
    using (SqlCommand com = new SqlCommand(createQuery, scon))
    {
        scon.Open();
        com.CommandText = dropQuery;
        com.ExecuteNonQuery();
        com.CommandText = createQuery;
        com.ExecuteNonQuery();

        string path =
@"C:\Users\user\source\repos\ITE365Assign7_2\players.txt";
        FileInfo DBfile = new FileInfo(path);
        if (DBfile.Exists)
        {
            StreamReader reader = new StreamReader(path);
            string tempBA = "";
            string tempN = "";
            string line = "", tempID = "";
            bool first = false, foundIt = false;
            int x = 0, ss = 0;
            while ((line = reader.ReadLine()) != null)
            {
                tempBA = ""; tempN = ""; foundIt = false; tempID = "";
                ss = 0;
                for (; ss < line.Length; ss++)
                {
                    if (line[ss] == ' ') { break; }
                    tempID += line[ss];
                }
                id.Add(tempID);

                for (x = ss; x < line.Length; x++)
                {
                    if (line[x] == ' ' && x < line.Length - 1)
                    {
                        if (char.IsLetter(line[x + 1]))
                        {
                            if (first) { first = false; }
                            else first = true;

                            tempN = "";
                            for (int z = x + 1; z < line.Length; z++)
                            {
                                tempN += line[z];
                                if (line[z] == ' ') { break; }
                            }
                            if (first) { fn.Add(tempN); }
                            else if (!first) { ln.Add(tempN); }
                        }
                    }
                }
            }
        }
    }
}

```

```

        if (char.IsDigit(line[x]) && foundIt == false)
        {
            foundIt = true;
            tempBA = "";
            for (int z = x; z < line.Length; z++)
            {
                tempBA += line[z];
            }
            ba.Add(Convert.ToDouble(tempBA));
        }
    }

    string insertQ = "";
    for (int rr = 0; rr < id.Count; rr++)
    {
        insertQ += "INSERT INTO
Player(PlayerID,FirstName,LastName,BattingAverage) " +
        "VALUES ('" + id[rr] + "','" + fn[rr] + "','" +
        + ln[rr] + "','" + ba[rr] + "');"
    }
    com.CommandText = insertQ;
    com.ExecuteNonQuery();
}
}
scon.Close();
}
}
protected void search_Click(object sender, EventArgs e)
{
    DataTable dt = new DataTable();
    dt.Columns.AddRange(new DataColumn[4] { new DataColumn("Player Id",
typeof(string)),
        new DataColumn("First Name", typeof(string)),
        new DataColumn("Last Name", typeof(string)),
        new DataColumn("Batting Average", typeof(float)) });
    StringBuilder sb = new StringBuilder();
    sb.Append("<table cellpadding='5' cellspacing='0' style='border: 1px solid
#ccc;' +
        "font-size: 9pt;font-family:Arial'>");

    using (SqlConnection con = new SqlConnection(connectString))
    {
        string searchQuery = "SELECT * FROM Player WHERE ";
        using (SqlCommand com = new SqlCommand(searchQuery, con))
        {
            con.Open();
            string q = String.Format("{0}", Request.Form["query"]);
            searchQuery += q;
            com.CommandText = searchQuery;
            //int row = 0;
            using (System.Data.SqlClient.SqlDataReader reader =
com.ExecuteReader())
            {
                while (reader.Read())
                {
                    dt.Rows.Add(reader["PlayerID"], reader["FirstName"],
                        reader["LastName"], reader["BattingAverage"]);
                }
            }
        }
    }
}

```



```

        found = true;
    }
    reader.Close();
}
}
con.Close();
}
sb.Append("<tr>");
foreach (DataColumn column in dt.Columns)
{
    sb.Append("<th style='border: 1px solid #ccc'>"
        + column.ColumnName + "</th>");
}
sb.Append("</tr>");

foreach (DataRow row in dt.Rows)
{
    sb.Append("<tr>");
    foreach (DataColumn column in dt.Columns)
    {
        sb.Append("<td style='width:100px;border: 1px solid #ccc'>"
            + row[column.ColumnName].ToString() + "</td>");
    }
    sb.Append("</tr>");
}
sb.Append("</table>");
table1.Text = sb.ToString();
if (!found)
{
    Response.Write("<script>alert('No Results Found');</script>");
}
}
}
}
}

```

Query: BattingAverage=0.21

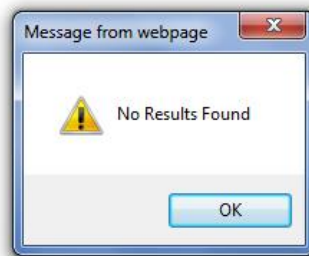
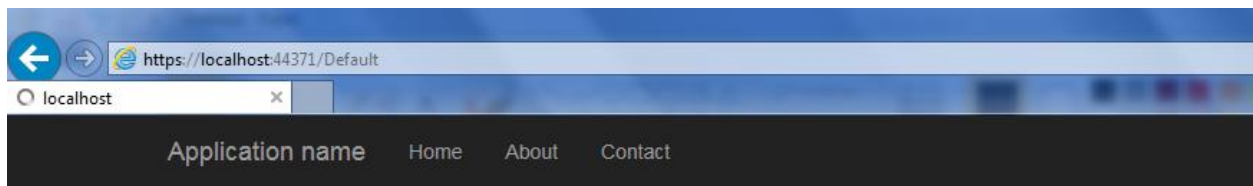
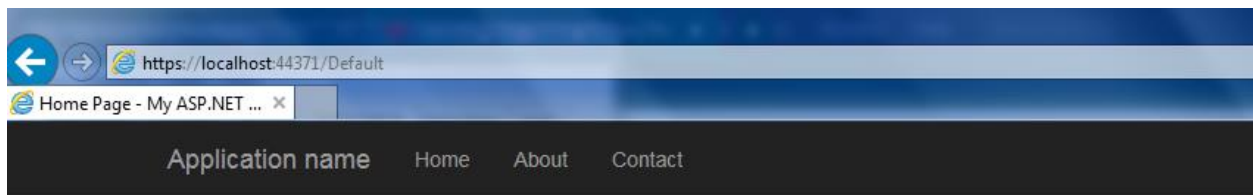


Application name Home About Contact

Query

Player Id	First Name	Last Name	Batting Average
322	John	Doe	0.21
561	Sally	Smith	0.21

LastName='Smith'



Sources

<https://www.aspsnippets.com/Articles/Create-HTML-Table-in-Code-Behind-in-ASPNet-using-C-and-VBNet.aspx>