

```

1 /*-----
2  * Author:      Dan Cassidy
3  * Date:        2015-06-23
4  * Assignment:  cView-P4
5  * Source File: Main.aspx.cs
6  * Language:    C#
7  * Course:      CSCI-C 490, C# Programming, MoWe 08:00
8  * Purpose:     Code-behind file for Main.aspx. Controls randomization and resetting of the tables
9  *              behind the application.
10 -----*/
11
12 using System;
13 using System.Collections.Generic;
14 using System.Linq;
15 using System.Web;
16 using System.Web.UI;
17 using System.Web.UI.WebControls;
18
19 namespace cView_P4_DanCassidy
20 {
21     public partial class Menu : System.Web.UI.Page
22     {
23         /*-----
24          * Name:      btnRandomize_Click
25          * Type:      Event Handler Method
26          * Purpose:   Handles randomizing the tables when clicked.
27          * Input:     object sender, holds a reference to the object that raised this event.
28          * Input:     EventArgs e, holds data related to this event.
29          * Output:    Nothing.
30          -----*/
31         protected void btnRandomize_Click(object sender, EventArgs e)
32         {
33             // Hide things until needed.
34             lblResult.Visible = false;
35             lblError.Visible = false;
36
37             try
38             {
39                 using (CViewDataEntities db = new CViewDataEntities())
40                 {
41                     // Used SQL statements because I didn't want to add more tables to the Entity
42                     // Framework model just for this. Also, it's easier.
43                     db.Database.ExecuteSqlCommand("TRUNCATE TABLE dbo.Business");
44                     db.Database.ExecuteSqlCommand("TRUNCATE TABLE dbo.BusinessReset");
45                     db.Database.ExecuteSqlCommand("TRUNCATE TABLE dbo.Park");
46                     db.Database.ExecuteSqlCommand("TRUNCATE TABLE dbo.ParkReset");
47                     db.Database.ExecuteSqlCommand("TRUNCATE TABLE dbo.PublicFacility");
48                     db.Database.ExecuteSqlCommand("TRUNCATE TABLE dbo.PublicFacilityReset");
49
50                     db.Database.ExecuteSqlCommand("INSERT INTO dbo.BusinessReset SELECT " +
51                         "TOP 50 * FROM dbo.BusinessBase ORDER BY NEWID()");
52                     db.Database.ExecuteSqlCommand("INSERT INTO dbo.Business SELECT " +
53                         " * FROM dbo.BusinessReset");
54                     db.Database.ExecuteSqlCommand("INSERT INTO dbo.ParkReset SELECT " +
55                         "TOP 50 * FROM dbo.ParkBase ORDER BY NEWID()");
56                     db.Database.ExecuteSqlCommand("INSERT INTO dbo.Park SELECT " +
57                         " * FROM dbo.ParkReset");
58                     db.Database.ExecuteSqlCommand("INSERT INTO dbo.PublicFacilityReset SELECT " +
59                         "TOP 50 * FROM dbo.PublicFacilityBase ORDER BY NEWID()");
60                     db.Database.ExecuteSqlCommand("INSERT INTO dbo.PublicFacility SELECT " +
61                         " * FROM dbo.PublicFacilityReset");
62                 }
63
64                 lblResult.Text = "The tables have been randomized.";
65                 lblResult.Visible = true;
66             }
67         }
68     }
69 }

```

```

67         catch
68         {
69             lblError.Text = "Error: Could not randomize the tables.";
70             lblError.Visible = true;
71         }
72     }
73
74     /*-----
75     * Name:      btnReset_Click
76     * Type:      Event Handler Method
77     * Purpose:   Handles resetting the tables to their prior randomized states.
78     * Input:    object sender, holds a reference to the object that raised this event.
79     * Input:    EventArgs e, holds data related to this event.
80     * Output:   Nothing.
81     -----*/
82     protected void btnReset_Click(object sender, EventArgs e)
83     {
84         // Hide things until needed.
85         lblResult.Visible = false;
86         lblError.Visible = false;
87
88         try
89         {
90             using (CViewDataEntities db = new CViewDataEntities())
91             {
92                 // Again, used SQL statements because I didn't want to add more tables to the
93                 // Entity Framework model just for this. Also, it's easier.
94                 db.Database.ExecuteSqlCommand("TRUNCATE TABLE dbo.Business");
95                 db.Database.ExecuteSqlCommand("TRUNCATE TABLE dbo.Park");
96                 db.Database.ExecuteSqlCommand("TRUNCATE TABLE dbo.PublicFacility");
97
98                 db.Database.ExecuteSqlCommand("INSERT INTO dbo.Business SELECT " +
99                     " * FROM dbo.BusinessReset");
100                db.Database.ExecuteSqlCommand("INSERT INTO dbo.Park SELECT " +
101                    " * FROM dbo.ParkReset");
102                db.Database.ExecuteSqlCommand("INSERT INTO dbo.PublicFacility SELECT " +
103                    " * FROM dbo.PublicFacilityReset");
104            }
105
106            lblResult.Text = "The tables have been reset to their prior randomized states.";
107            lblResult.Visible = true;
108        }
109        catch
110        {
111            lblError.Text = "Error: Could not reset the tables.";
112            lblError.Visible = true;
113        }
114    }
115 }
116 }

```