using System;

using System.Collections.Generic;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Configuration;

public partial class MemberPages\_QuerySiteSearch : System.Web.UI.Page

{

public class foo

{

//for site id

public static string Site\_ID;

//check for valid site

public static string checksite = "";

public static int ispopulated;

//for dropdowns

public static string[] State = new string[100];

public static string[] City = new string[2000];

public static string[] SiteName = new string[20000];

public static string[] Market = new string[2000];

public static string[] SiteID = new string[20000];

public static string State1;

public static string City1;

public static string SiteName1;

public static string Market1;

public static string SiteID1;

}

//Classes to hold info

private Table\_Info generalInfo;

private RFTable\_Info rftableInfo;

private Table\_Info ancillary;

private tabDone tabdone;

private SiteNotes notes;

private Audit21Login login;

public bool TimeoutControlEnabled

{

get { return Timeout1.Enabled; }

set { Timeout1.Enabled = value; }

}

protected void Page\_Load(object sender, EventArgs e)

{

//Label1.Text = String.Format("Timeout Warning Popup will display in {0} minute(s).", HttpContext.Current.Session.Timeout - 1);

Timeout1.TimeoutMinutes = HttpContext.Current.Session.Timeout;

Timeout1.AboutToTimeoutMinutes = HttpContext.Current.Session.Timeout - 1;

//Session.Abandon();

// if (!IsPostBack)

// {

generalInfo = (Table\_Info)Session["Table\_Info1"];

rftableInfo = (RFTable\_Info)Session["Table\_Info2"];

ancillary = (Table\_Info)Session["Table\_Info3"];

tabdone = (tabDone)Session["tabdone1"];

notes = (SiteNotes)Session["Notes"];

login = (Audit21Login)Session["login"];

// }

if (tabdone == null)

{

tabdone = new tabDone();

}

if (login == null)

{

login = new Audit21Login();

Response.Redirect("~/Login.aspx");

}

if (login != null)

{

if (login.UserName.Length != 0)

{

lbUserName.Text = login.UserName;

if (!Page.IsPostBack)

{

findStates();

findCities();

findSiteNames();

findMarkets();

findSiteIDs();

}

if (String.Compare(login.role, "administrator") == 0)

{

btnAdmin.Enabled = true;

btnAdmin.Visible = true;

}

}

else

{

Response.Redirect("~/Login.aspx");

}

}

if (Page.IsPostBack)

{

foo.State1 = ddlState.SelectedItem.ToString();

foo.City1 = ddlCity.SelectedItem.ToString();

foo.SiteName1 = ddlSiteName.SelectedItem.ToString();

foo.Market1 = ddlMarket.SelectedItem.ToString();

foo.SiteID1 = ddlSiteID.SelectedItem.ToString();

}

}

protected void btnGetSiteInfo\_Click(object sender, EventArgs e)

{

foo.checksite = "";

foo.Site\_ID = ddlSiteID.SelectedItem.ToString();

PullAllData();

string url = "";

string SiteKey = ddlSiteID.SelectedItem.ToString();

if (string.Compare(foo.Site\_ID, "--Choose--") == 0)

{

lbmessage.ForeColor = System.Drawing.Color.Red;

lbmessage.Text = "Please choose a site ID to continue";

}

else

{

Session["tabdone1"] = tabdone;

url = "~/MemberPages/AddGeneralInfo.aspx?";

url += "SiteKey=" + SiteKey;

Response.Redirect(url);

}

}

protected void btnGetCarrier\_Click(object sender, ImageClickEventArgs e)

{

foo.checksite = "";

foo.Site\_ID = ddlSiteID.SelectedItem.ToString();

PullAllData();

string url = "";

string SiteKey = ddlSiteID.SelectedItem.ToString();

if (string.Compare(foo.Site\_ID, "--Choose--") == 0)

{

lbmessage.ForeColor = System.Drawing.Color.Red;

lbmessage.Text = "Please choose a site ID to continue";

}

else

{

Session["tabdone1"] = tabdone;

url = "~/MemberPages/AddGeneralInfo.aspx?";

url += "SiteKey=" + SiteKey;

Response.Redirect(url);

}

}

//protected void btnGetCarrier\_Click(object sender, ImageClickEventArgs e)

//{

// foo.checksite = "";

// foo.Site\_ID = ddlSiteID.SelectedItem.ToString();

// PullAllData();

// string url = "";

// string SiteKey = ddlSiteID.SelectedItem.ToString();

// if (string.Compare(foo.Site\_ID, "--Choose--") == 0)

// {

// lbmessage.ForeColor = System.Drawing.Color.Red;

// lbmessage.Text = "Please choose a site ID to continue";

// }

// else

// {

// Session["tabdone1"] = tabdone;

// url = "~/MemberPages/AddGeneralInfo.aspx?";

// url += "SiteKey=" + SiteKey;

// Response.Redirect(url);

// }

//}

protected void changedState\_OnSelectedIndexChanged(object sender, EventArgs e)

{

if (string.Compare(foo.State1, "--Choose--") == 0)

{

}

else

{

//Alter cities dropdown

ddlCity.Items.Clear();

foo.City[0] = "--Choose--";

for (int w = 1; w < 2000; w++)

{

foo.City[w] = "";

}

int i = 1;

//foo.State1 = ddlState.SelectedItem.ToString();

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT CITY " +

"FROM SITE\_INFORMATION1 " +

"WHERE STATE = @STATE ORDER BY CITY ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@STATE", foo.State1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.City[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlCity.Items.Add(foo.City[c]);

}

//Alter site names dropdown

ddlSiteName.Items.Clear();

foo.SiteName[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteName[w] = "";

}

i = 1;

cmd = null;

queryString = @"SELECT DISTINCT SITE\_NAME " +

"FROM SITE\_INFORMATION1 " +

"WHERE STATE = @STATE ORDER BY SITE\_NAME ASC";

//queryString = @"SELECT DISTINCT SITE\_NAME " +

// "FROM SITE\_INFORMATION1 " +

// "WHERE CITY = @CITY ORDER BY SITE\_NAME ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@STATE", foo.State1);

//cmd.Parameters.AddWithValue("@CITY", foo.City1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteName[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteName.Items.Add(foo.SiteName[c]);

}

//Alter markets dropdown

ddlMarket.Items.Clear();

foo.Market[0] = "--Choose--";

for (int w = 1; w < 2000; w++)

{

foo.Market[w] = "";

}

i = 1;

cmd = null;

queryString = @"SELECT DISTINCT OI.MARKET " +

"FROM SITE\_INFORMATION1 SI LEFT OUTER JOIN OPERATOR\_INFORMATION OI " +

"ON SI.SITE\_ID = OI.SITE\_ID WHERE LEN(SI.SITE\_ID) > 0 AND SI.STATE = @STATE ORDER BY OI.MARKET ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@STATE", foo.State1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.Market[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlMarket.Items.Add(foo.Market[c]);

}

//Alter site ids dropdown

ddlSiteID.Items.Clear();

foo.SiteID[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteID[w] = "";

}

i = 1;

cmd = null;

queryString = @"SELECT DISTINCT SITE\_ID " +

"FROM SITE\_INFORMATION1 " +

"WHERE LEN(SITE\_ID) > 0 AND STATE = @STATE ORDER BY SITE\_ID ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@STATE", foo.State1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteID[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteID.Items.Add(foo.SiteID[c]);

}

}

}

protected void changedCity\_OnSelectedIndexChanged(object sender, EventArgs e)

{

if (string.Compare(foo.City1, "--Choose--") == 1)

{

//Alter site names dropdown

ddlSiteName.Items.Clear();

foo.SiteName[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteName[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT SITE\_NAME " +

"FROM SITE\_INFORMATION1 " +

"WHERE STATE = @STATE AND CITY = @CITY ORDER BY SITE\_NAME ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@STATE", foo.State1);

cmd.Parameters.AddWithValue("@CITY", foo.City1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteName[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteName.Items.Add(foo.SiteName[c]);

}

//Alter markets dropdown

ddlMarket.Items.Clear();

foo.Market[0] = "--Choose--";

for (int w = 1; w < 2000; w++)

{

foo.Market[w] = "";

}

i = 1;

cmd = null;

queryString = @"SELECT DISTINCT OI.MARKET " +

"FROM SITE\_INFORMATION1 SI LEFT OUTER JOIN OPERATOR\_INFORMATION OI " +

"ON SI.SITE\_ID = OI.SITE\_ID WHERE LEN(SI.SITE\_ID) > 0 AND SI.STATE = @STATE AND SI.CITY = @CITY ORDER BY OI.MARKET ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@STATE", foo.State1);

cmd.Parameters.AddWithValue("@CITY", foo.City1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.Market[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlMarket.Items.Add(foo.Market[c]);

}

//Alter site ids dropdown

ddlSiteID.Items.Clear();

foo.SiteID[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteID[w] = "";

}

i = 1;

cmd = null;

queryString = @"SELECT DISTINCT SITE\_ID " +

"FROM SITE\_INFORMATION1 " +

"WHERE LEN(SITE\_ID) > 0 AND STATE = @STATE AND CITY = @CITY ORDER BY SITE\_ID ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@STATE", foo.State1);

cmd.Parameters.AddWithValue("@CITY", foo.City1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteID[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteID.Items.Add(foo.SiteID[c]);

}

}

else if (string.Compare(foo.State1, "--Choose--") == 0 & string.Compare(foo.City1, "--Choose--") == 1)

{

//Alter site names dropdown

ddlSiteName.Items.Clear();

foo.SiteName[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteName[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT SITE\_NAME " +

"FROM SITE\_INFORMATION1 " +

"WHERE CITY = @CITY ORDER BY SITE\_NAME ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@CITY", foo.City1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteName[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteName.Items.Add(foo.SiteName[c]);

}

//Alter markets dropdown

ddlMarket.Items.Clear();

foo.Market[0] = "--Choose--";

for (int w = 1; w < 2000; w++)

{

foo.Market[w] = "";

}

i = 1;

cmd = null;

queryString = @"SELECT DISTINCT OI.MARKET " +

"FROM SITE\_INFORMATION1 SI LEFT OUTER JOIN OPERATOR\_INFORMATION OI " +

"ON SI.SITE\_ID = OI.SITE\_ID WHERE LEN(SI.SITE\_ID) > 0 AND SI.CITY = @CITY ORDER BY OI.MARKET ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@CITY", foo.City1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.Market[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlMarket.Items.Add(foo.Market[c]);

}

//Alter site ids dropdown

ddlSiteID.Items.Clear();

foo.SiteID[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteID[w] = "";

}

i = 1;

cmd = null;

queryString = @"SELECT DISTINCT SITE\_ID " +

"FROM SITE\_INFORMATION1 " +

"WHERE LEN(SITE\_ID) > 0 AND CITY = @CITY ORDER BY SITE\_ID ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@CITY", foo.City1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteID[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteID.Items.Add(foo.SiteID[c]);

}

}

else

{

}

}

protected void changedSiteName\_OnSelectedIndexChanged(object sender, EventArgs e)

{

if (string.Compare(foo.SiteName1, "--Choose--") == 0)

{

}

else if (string.Compare(foo.State1, "--Choose--") == 1 & string.Compare(foo.City1, "--Choose--") == 1 & string.Compare(foo.SiteName1, "--Choose--") == 1)

{

//Alter markets dropdown

ddlMarket.Items.Clear();

foo.Market[0] = "--Choose--";

for (int w = 1; w < 2000; w++)

{

foo.Market[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT OI.MARKET " +

"FROM SITE\_INFORMATION1 SI LEFT OUTER JOIN OPERATOR\_INFORMATION OI " +

"ON SI.SITE\_ID = OI.SITE\_ID WHERE LEN(SI.SITE\_ID) > 0 AND SI.STATE = @STATE AND SI.CITY = @CITY AND SI.SITE\_NAME = @SITE\_NAME ORDER BY OI.MARKET ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@STATE", foo.State1);

cmd.Parameters.AddWithValue("@CITY", foo.City1);

cmd.Parameters.AddWithValue("@SITE\_NAME", foo.SiteName1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.Market[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlMarket.Items.Add(foo.Market[c]);

}

//Alter site ids dropdown

ddlSiteID.Items.Clear();

foo.SiteID[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteID[w] = "";

}

i = 1;

cmd = null;

queryString = @"SELECT DISTINCT SITE\_ID " +

"FROM SITE\_INFORMATION1 " +

"WHERE LEN(SITE\_ID) > 0 AND STATE = @STATE AND CITY = @CITY AND SITE\_NAME = @SITE\_NAME ORDER BY SITE\_ID ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@STATE", foo.State1);

cmd.Parameters.AddWithValue("@CITY", foo.City1);

cmd.Parameters.AddWithValue("@SITE\_NAME", foo.SiteName1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteID[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteID.Items.Add(foo.SiteID[c]);

}

}

else if (string.Compare(foo.State1, "--Choose--") == 0 & string.Compare(foo.City1, "--Choose--") == 1 & string.Compare(foo.SiteName1, "--Choose--") == 1)

{

//Alter markets dropdown

ddlMarket.Items.Clear();

foo.Market[0] = "--Choose--";

for (int w = 1; w < 2000; w++)

{

foo.Market[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT OI.MARKET " +

"FROM SITE\_INFORMATION1 SI LEFT OUTER JOIN OPERATOR\_INFORMATION OI " +

"ON SI.SITE\_ID = OI.SITE\_ID WHERE LEN(SI.SITE\_ID) > 0 AND SI.CITY = @CITY AND SI.SITE\_NAME = @SITE\_NAME ORDER BY OI.MARKET ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@CITY", foo.City1);

cmd.Parameters.AddWithValue("@SITE\_NAME", foo.SiteName1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.Market[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlMarket.Items.Add(foo.Market[c]);

}

//Alter site ids dropdown

ddlSiteID.Items.Clear();

foo.SiteID[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteID[w] = "";

}

i = 1;

cmd = null;

queryString = @"SELECT DISTINCT SITE\_ID " +

"FROM SITE\_INFORMATION1 " +

"WHERE LEN(SITE\_ID) > 0 AND CITY = @CITY AND SITE\_NAME = @SITE\_NAME ORDER BY SITE\_ID ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@CITY", foo.City1);

cmd.Parameters.AddWithValue("@SITE\_NAME", foo.SiteName1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteID[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteID.Items.Add(foo.SiteID[c]);

}

}

else if (string.Compare(foo.State1, "--Choose--") == 0 & string.Compare(foo.City1, "--Choose--") == 0 & string.Compare(foo.SiteName1, "--Choose--") == 1)

{

//Alter markets dropdown

ddlMarket.Items.Clear();

foo.Market[0] = "--Choose--";

for (int w = 1; w < 2000; w++)

{

foo.Market[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT OI.MARKET " +

"FROM SITE\_INFORMATION1 SI LEFT OUTER JOIN OPERATOR\_INFORMATION OI " +

"ON SI.SITE\_ID = OI.SITE\_ID WHERE LEN(SI.SITE\_ID) > 0 AND SI.SITE\_NAME = @SITE\_NAME ORDER BY OI.MARKET ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@SITE\_NAME", foo.SiteName1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.Market[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlMarket.Items.Add(foo.Market[c]);

}

//Alter site ids dropdown

ddlSiteID.Items.Clear();

foo.SiteID[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteID[w] = "";

}

i = 1;

cmd = null;

queryString = @"SELECT DISTINCT SITE\_ID " +

"FROM SITE\_INFORMATION1 " +

"WHERE LEN(SITE\_ID) > 0 AND SITE\_NAME = @SITE\_NAME ORDER BY SITE\_ID ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@SITE\_NAME", foo.SiteName1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteID[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteID.Items.Add(foo.SiteID[c]);

}

}

else if (string.Compare(foo.State1, "--Choose--") == 1 & string.Compare(foo.City1, "--Choose--") == 0 & string.Compare(foo.SiteName1, "--Choose--") == 1)

{

//Alter markets dropdown

ddlMarket.Items.Clear();

foo.Market[0] = "--Choose--";

for (int w = 1; w < 2000; w++)

{

foo.Market[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT OI.MARKET " +

"FROM SITE\_INFORMATION1 SI LEFT OUTER JOIN OPERATOR\_INFORMATION OI " +

"ON SI.SITE\_ID = OI.SITE\_ID WHERE LEN(SI.SITE\_ID) > 0 AND SI.STATE = @STATE AND SI.SITE\_NAME = @SITE\_NAME ORDER BY OI.MARKET ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@STATE", foo.State1);

cmd.Parameters.AddWithValue("@SITE\_NAME", foo.SiteName1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.Market[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlMarket.Items.Add(foo.Market[c]);

}

//Alter site ids dropdown

ddlSiteID.Items.Clear();

foo.SiteID[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteID[w] = "";

}

i = 1;

cmd = null;

queryString = @"SELECT DISTINCT SITE\_ID " +

"FROM SITE\_INFORMATION1 " +

"WHERE LEN(SITE\_ID) > 0 AND STATE = @STATE AND SITE\_NAME = @SITE\_NAME ORDER BY SITE\_ID ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@STATE", foo.State1);

cmd.Parameters.AddWithValue("@SITE\_NAME", foo.SiteName1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteID[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteID.Items.Add(foo.SiteID[c]);

}

}

//TESTING CODE

//if (string.Compare(foo.City1, "--Choose--") == 1 & string.Compare(foo.SiteName1, "--Choose--") == 1)

//{

// //Alter site names dropdown

// ddlSiteName.Items.Clear();

// foo.SiteName[0] = "--Choose--";

// for (int w = 1; w < 20000; w++)

// {

// foo.SiteName[w] = "";

// }

// int i = 1;

// SqlCommand cmd = null;

// string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

// string queryString = @"SELECT DISTINCT SITE\_NAME " +

// "FROM SITE\_INFORMATION1 " +

// "WHERE CITY = @CITY AND SITE\_NAME = @SITE\_NAME ORDER BY SITE\_NAME ASC";

// using (SqlConnection connection =

// new SqlConnection(connectionString))

// {

// SqlCommand command =

// new SqlCommand(queryString, connection);

// connection.Open();

// cmd = new SqlCommand(queryString);

// cmd.Connection = connection;

// cmd.Parameters.AddWithValue("@CITY", foo.City1);

// cmd.Parameters.AddWithValue("@SITE\_NAME", foo.SiteName1);

// SqlDataReader reader = cmd.ExecuteReader();

// // Call Read before accessing data.

// while (reader.Read())

// {

// foo.SiteName[i] = reader[0].ToString();

// i++;

// }

// // Call Close when done reading.

// reader.Close();

// }

// for (int c = 0; c < i; c++)

// {

// ddlSiteName.Items.Add(foo.SiteName[c]);

// }

// //Alter markets dropdown

// ddlMarket.Items.Clear();

// foo.Market[0] = "--Choose--";

// for (int w = 1; w < 2000; w++)

// {

// foo.Market[w] = "";

// }

// i = 1;

// cmd = null;

// queryString = @"SELECT DISTINCT OI.MARKET " +

// "FROM SITE\_INFORMATION1 SI LEFT OUTER JOIN OPERATOR\_INFORMATION OI " +

// "ON SI.SITE\_ID = OI.SITE\_ID WHERE LEN(SI.SITE\_ID) > 0 AND SI.CITY = @CITY AND SI.SITE\_NAME = @SITE\_NAME ORDER BY OI.MARKET ASC";

// using (SqlConnection connection =

// new SqlConnection(connectionString))

// {

// SqlCommand command =

// new SqlCommand(queryString, connection);

// connection.Open();

// cmd = new SqlCommand(queryString);

// cmd.Connection = connection;

// cmd.Parameters.AddWithValue("@CITY", foo.City1);

// cmd.Parameters.AddWithValue("@SITE\_NAME", foo.SiteName1);

// SqlDataReader reader = cmd.ExecuteReader();

// // Call Read before accessing data.

// while (reader.Read())

// {

// foo.Market[i] = reader[0].ToString();

// i++;

// }

// // Call Close when done reading.

// reader.Close();

// }

// for (int c = 0; c < i; c++)

// {

// ddlMarket.Items.Add(foo.Market[c]);

// }

// //Alter site ids dropdown

// ddlSiteID.Items.Clear();

// foo.SiteID[0] = "--Choose--";

// for (int w = 1; w < 20000; w++)

// {

// foo.SiteID[w] = "";

// }

// i = 1;

// cmd = null;

// queryString = @"SELECT DISTINCT SITE\_ID " +

// "FROM SITE\_INFORMATION1 " +

// "WHERE LEN(SITE\_ID) > 0 AND CITY = @CITY AND SITE\_NAME = @SITE\_NAME ORDER BY SITE\_ID ASC";

// using (SqlConnection connection =

// new SqlConnection(connectionString))

// {

// SqlCommand command =

// new SqlCommand(queryString, connection);

// connection.Open();

// cmd = new SqlCommand(queryString);

// cmd.Connection = connection;

// cmd.Parameters.AddWithValue("@CITY", foo.City1);

// cmd.Parameters.AddWithValue("@SITE\_NAME", foo.SiteName1);//STOPED HERE

// SqlDataReader reader = cmd.ExecuteReader();

// // Call Read before accessing data.

// while (reader.Read())

// {

// foo.SiteID[i] = reader[0].ToString();

// i++;

// }

// // Call Close when done reading.

// reader.Close();

// }

// for (int c = 0; c < i; c++)

// {

// ddlSiteID.Items.Add(foo.SiteID[c]);

// }

//}

//else if (string.Compare(foo.State1, "--Choose--") == 0 & string.Compare(foo.City1, "--Choose--") == 1)

//{

// //Alter site names dropdown

// ddlSiteName.Items.Clear();

// foo.SiteName[0] = "--Choose--";

// for (int w = 1; w < 20000; w++)

// {

// foo.SiteName[w] = "";

// }

// int i = 1;

// SqlCommand cmd = null;

// string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

// string queryString = @"SELECT DISTINCT SITE\_NAME " +

// "FROM SITE\_INFORMATION1 " +

// "WHERE CITY = @CITY ORDER BY SITE\_NAME ASC";

// using (SqlConnection connection =

// new SqlConnection(connectionString))

// {

// SqlCommand command =

// new SqlCommand(queryString, connection);

// connection.Open();

// cmd = new SqlCommand(queryString);

// cmd.Connection = connection;

// cmd.Parameters.AddWithValue("@CITY", foo.City1);

// SqlDataReader reader = cmd.ExecuteReader();

// // Call Read before accessing data.

// while (reader.Read())

// {

// foo.SiteName[i] = reader[0].ToString();

// i++;

// }

// // Call Close when done reading.

// reader.Close();

// }

// for (int c = 0; c < i; c++)

// {

// ddlSiteName.Items.Add(foo.SiteName[c]);

// }

// //Alter markets dropdown

// ddlMarket.Items.Clear();

// foo.Market[0] = "--Choose--";

// for (int w = 1; w < 2000; w++)

// {

// foo.Market[w] = "";

// }

// i = 1;

// cmd = null;

// queryString = @"SELECT DISTINCT OI.MARKET " +

// "FROM SITE\_INFORMATION1 SI LEFT OUTER JOIN OPERATOR\_INFORMATION OI " +

// "ON SI.SITE\_ID = OI.SITE\_ID WHERE LEN(SI.SITE\_ID) > 0 AND SI.CITY = @CITY ORDER BY OI.MARKET ASC";

// using (SqlConnection connection =

// new SqlConnection(connectionString))

// {

// SqlCommand command =

// new SqlCommand(queryString, connection);

// connection.Open();

// cmd = new SqlCommand(queryString);

// cmd.Connection = connection;

// cmd.Parameters.AddWithValue("@CITY", foo.City1);

// SqlDataReader reader = cmd.ExecuteReader();

// // Call Read before accessing data.

// while (reader.Read())

// {

// foo.Market[i] = reader[0].ToString();

// i++;

// }

// // Call Close when done reading.

// reader.Close();

// }

// for (int c = 0; c < i; c++)

// {

// ddlMarket.Items.Add(foo.Market[c]);

// }

// //Alter site ids dropdown

// ddlSiteID.Items.Clear();

// foo.SiteID[0] = "--Choose--";

// for (int w = 1; w < 20000; w++)

// {

// foo.SiteID[w] = "";

// }

// i = 1;

// cmd = null;

// queryString = @"SELECT DISTINCT SITE\_ID " +

// "FROM SITE\_INFORMATION1 " +

// "WHERE LEN(SITE\_ID) > 0 AND CITY = @CITY ORDER BY SITE\_ID ASC";

// using (SqlConnection connection =

// new SqlConnection(connectionString))

// {

// SqlCommand command =

// new SqlCommand(queryString, connection);

// connection.Open();

// cmd = new SqlCommand(queryString);

// cmd.Connection = connection;

// cmd.Parameters.AddWithValue("@CITY", foo.City1);

// SqlDataReader reader = cmd.ExecuteReader();

// // Call Read before accessing data.

// while (reader.Read())

// {

// foo.SiteID[i] = reader[0].ToString();

// i++;

// }

// // Call Close when done reading.

// reader.Close();

// }

// for (int c = 0; c < i; c++)

// {

// ddlSiteID.Items.Add(foo.SiteID[c]);

// }

//}

//else

//{

//}

}

protected void changedMarket\_OnSelectedIndexChanged(object sender, EventArgs e)

{

if (string.Compare(foo.Market1, "--Choose--") == 0)

{

}

else if (string.Compare(foo.State1, "--Choose--") == 1 & string.Compare(foo.City1, "--Choose--") == 1 & string.Compare(foo.SiteName1, "--Choose--") == 1)

{

//Alter markets dropdown

ddlSiteID.Items.Clear();

foo.SiteID[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteID[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT SI.SITE\_ID " +

"FROM SITE\_INFORMATION1 SI LEFT OUTER JOIN OPERATOR\_INFORMATION OI " +

"ON SI.SITE\_ID = OI.SITE\_ID WHERE LEN(SI.SITE\_ID) > 0 AND SI.STATE = @STATE AND SI.CITY = @CITY AND SI.SITE\_NAME = @SITE\_NAME AND OI.MARKET = @MARKET ORDER BY SI.SITE\_ID ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@STATE", foo.State1);

cmd.Parameters.AddWithValue("@CITY", foo.City1);

cmd.Parameters.AddWithValue("@SITE\_NAME", foo.SiteName1);

cmd.Parameters.AddWithValue("@MARKET", foo.Market1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteID[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteID.Items.Add(foo.SiteID[c]);

}

}

else if (string.Compare(foo.State1, "--Choose--") == 1 & string.Compare(foo.City1, "--Choose--") == 1 & string.Compare(foo.SiteName1, "--Choose--") == 0)

{

//Alter markets dropdown

ddlSiteID.Items.Clear();

foo.SiteID[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteID[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT SI.SITE\_ID " +

"FROM SITE\_INFORMATION1 SI LEFT OUTER JOIN OPERATOR\_INFORMATION OI " +

"ON SI.SITE\_ID = OI.SITE\_ID WHERE LEN(SI.SITE\_ID) > 0 AND SI.STATE = @STATE AND SI.CITY = @CITY AND OI.MARKET = @MARKET ORDER BY SI.SITE\_ID ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@STATE", foo.State1);

cmd.Parameters.AddWithValue("@CITY", foo.City1);

cmd.Parameters.AddWithValue("@MARKET", foo.Market1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteID[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteID.Items.Add(foo.SiteID[c]);

}

}

else if (string.Compare(foo.State1, "--Choose--") == 1 & string.Compare(foo.City1, "--Choose--") == 0 & string.Compare(foo.SiteName1, "--Choose--") == 1)

{

//Alter markets dropdown

ddlSiteID.Items.Clear();

foo.SiteID[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteID[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT SI.SITE\_ID " +

"FROM SITE\_INFORMATION1 SI LEFT OUTER JOIN OPERATOR\_INFORMATION OI " +

"ON SI.SITE\_ID = OI.SITE\_ID WHERE LEN(SI.SITE\_ID) > 0 AND SI.STATE = @STATE AND SI.SITE\_NAME = @SITE\_NAME AND OI.MARKET = @MARKET ORDER BY SI.SITE\_ID ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@STATE", foo.State1);

cmd.Parameters.AddWithValue("@SITE\_NAME", foo.SiteName1);

cmd.Parameters.AddWithValue("@MARKET", foo.Market1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteID[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteID.Items.Add(foo.SiteID[c]);

}

}

else if (string.Compare(foo.State1, "--Choose--") == 0 & string.Compare(foo.City1, "--Choose--") == 1 & string.Compare(foo.SiteName1, "--Choose--") == 1)

{

//Alter markets dropdown

ddlSiteID.Items.Clear();

foo.SiteID[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteID[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT SI.SITE\_ID " +

"FROM SITE\_INFORMATION1 SI LEFT OUTER JOIN OPERATOR\_INFORMATION OI " +

"ON SI.SITE\_ID = OI.SITE\_ID WHERE LEN(SI.SITE\_ID) > 0 AND SI.CITY = @CITY AND SI.SITE\_NAME = @SITE\_NAME AND OI.MARKET = @MARKET ORDER BY SI.SITE\_ID ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@CITY", foo.City1);

cmd.Parameters.AddWithValue("@SITE\_NAME", foo.SiteName1);

cmd.Parameters.AddWithValue("@MARKET", foo.Market1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteID[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteID.Items.Add(foo.SiteID[c]);

}

}

else if (string.Compare(foo.State1, "--Choose--") == 1 & string.Compare(foo.City1, "--Choose--") == 0 & string.Compare(foo.SiteName1, "--Choose--") == 0)

{

//Alter markets dropdown

ddlSiteID.Items.Clear();

foo.SiteID[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteID[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT SI.SITE\_ID " +

"FROM SITE\_INFORMATION1 SI LEFT OUTER JOIN OPERATOR\_INFORMATION OI " +

"ON SI.SITE\_ID = OI.SITE\_ID WHERE LEN(SI.SITE\_ID) > 0 AND SI.STATE = @STATE AND OI.MARKET = @MARKET ORDER BY SI.SITE\_ID ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@STATE", foo.State1);

cmd.Parameters.AddWithValue("@MARKET", foo.Market1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteID[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteID.Items.Add(foo.SiteID[c]);

}

}

else if (string.Compare(foo.State1, "--Choose--") == 0 & string.Compare(foo.City1, "--Choose--") == 0 & string.Compare(foo.SiteName1, "--Choose--") == 1)

{

//Alter markets dropdown

ddlSiteID.Items.Clear();

foo.SiteID[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteID[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT SI.SITE\_ID " +

"FROM SITE\_INFORMATION1 SI LEFT OUTER JOIN OPERATOR\_INFORMATION OI " +

"ON SI.SITE\_ID = OI.SITE\_ID WHERE LEN(SI.SITE\_ID) > 0 AND SI.SITE\_NAME = @SITE\_NAME AND OI.MARKET = @MARKET ORDER BY SI.SITE\_ID ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@SITE\_NAME", foo.SiteName1);

cmd.Parameters.AddWithValue("@MARKET", foo.Market1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteID[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteID.Items.Add(foo.SiteID[c]);

}

}

else if (string.Compare(foo.State1, "--Choose--") == 0 & string.Compare(foo.City1, "--Choose--") == 1 & string.Compare(foo.SiteName1, "--Choose--") == 0)

{

//Alter markets dropdown

ddlSiteID.Items.Clear();

foo.SiteID[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteID[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT SI.SITE\_ID " +

"FROM SITE\_INFORMATION1 SI LEFT OUTER JOIN OPERATOR\_INFORMATION OI " +

"ON SI.SITE\_ID = OI.SITE\_ID WHERE LEN(SI.SITE\_ID) > 0 AND SI.CITY = @CITY AND OI.MARKET = @MARKET ORDER BY SI.SITE\_ID ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@CITY", foo.City1);

cmd.Parameters.AddWithValue("@MARKET", foo.Market1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteID[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteID.Items.Add(foo.SiteID[c]);

}

}

else if (string.Compare(foo.State1, "--Choose--") == 0 & string.Compare(foo.City1, "--Choose--") == 0 & string.Compare(foo.SiteName1, "--Choose--") == 0)

{

//Alter markets dropdown

ddlSiteID.Items.Clear();

foo.SiteID[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteID[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT SI.SITE\_ID " +

"FROM SITE\_INFORMATION1 SI LEFT OUTER JOIN OPERATOR\_INFORMATION OI " +

"ON SI.SITE\_ID = OI.SITE\_ID WHERE LEN(SI.SITE\_ID) > 0 AND OI.MARKET = @MARKET ORDER BY SI.SITE\_ID ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.AddWithValue("@MARKET", foo.Market1);

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteID[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteID.Items.Add(foo.SiteID[c]);

}

}

}

//Pull all info for preinstall from database

private void PullAllData()

{

//Find everything in database for general tab and save to session

if (generalInfo == null)

{

generalInfo = new Table\_Info();

}

if (rftableInfo == null)

{

rftableInfo = new RFTable\_Info();

}

if (ancillary == null)

{

ancillary = new Table\_Info();

}

if (notes == null)

{

notes = new SiteNotes();

}

ClearFieldsForNewSearch();

GetOperatorInfo();

GetSiteInfo();

GetBTSInfo();

GetAdditionalInfo();

getGeneralTimeStamps();

Session["Table\_Info1"] = generalInfo;

//Find everything in database for rf tab and save to session

GetFrequencyInfo();

GetSectorInfo();

getRFTimeStamps();

Session["Table\_Info2"] = rftableInfo;

//Find everything in database for ancillary tab and save to session

GetAncillaryInfo();

getAncillaryTimeStamps();

Session["Table\_Info3"] = ancillary;

//Find everything in database for site notes tabe and save to session

GetSiteNotes();

getNotesTimeStamps();

Session["Notes"] = notes;

//for floorplans and photos tabs to be enables

GetFloorPlans();

GetPhotos();

}

private void ClearFieldsForNewSearch()

{

generalInfo.Carrier = "";

generalInfo.Market = "";

generalInfo.RFTechnology = "";

generalInfo.VendorPO = "";

generalInfo.SiteName = "";

generalInfo.Latitude = "";

generalInfo.Longitute = "";

generalInfo.SiteAddress = "";

generalInfo.City = "";

generalInfo.State = "";

generalInfo.SiteContact = "";

generalInfo.ContactEmail = "";

generalInfo.ContactPhone = "";

generalInfo.AccessCode = "";

generalInfo.AccessInstructions = "";

generalInfo.SiteID = "";

generalInfo.TempSiteID = "";

generalInfo.VendorType = "Lucent";

generalInfo.Model = "";

generalInfo.cabinetFrames = "1";

generalInfo.ExpansionFrames = "1";

generalInfo.CustomerSitePO = "";

generalInfo.NumberOfSectors = "1";

for (int stamps = 0; stamps < 50; stamps++)

{

generalInfo.generaltimestamp[stamps] = "";

generalInfo.usernamegeneral[stamps] = "";

generalInfo.unitnumbergeneral[stamps] = "";

rftableInfo.rftimestamp[stamps] = "";

rftableInfo.usernamerf[stamps] = "";

rftableInfo.unitnumberrf[stamps] = "";

ancillary.ancillarytimestamp[stamps] = "";

ancillary.usernameancillary[stamps] = "";

ancillary.unitnumberancillary[stamps] = "";

notes.notestimestamp[stamps] = "";

notes.usernamenotes[stamps] = "";

notes.unitnumbernotes[stamps] = "";

}

}

protected void findStates()

{

ddlState.Items.Clear();

foo.State[0] = "--Choose--";

for (int w = 1; w < 100; w++)

{

foo.State[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT STATE " +

"FROM SITE\_INFORMATION1 WHERE Len(STATE) > 0 " +

"ORDER BY STATE ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.State[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlState.Items.Add(foo.State[c]);

}

}

protected void findCities()

{

ddlCity.Items.Clear();

foo.City[0] = "--Choose--";

for (int w = 1; w < 2000; w++)

{

foo.City[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT CITY " +

"FROM SITE\_INFORMATION1 WHERE Len(CITY) > 0 " +

"ORDER BY CITY ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.City[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlCity.Items.Add(foo.City[c]);

}

}

protected void findSiteNames()

{

ddlSiteName.Items.Clear();

foo.SiteName[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteName[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT SITE\_NAME " +

"FROM SITE\_INFORMATION1 WHERE Len(SITE\_NAME) > 0 " +

"ORDER BY SITE\_NAME ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteName[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteName.Items.Add(foo.SiteName[c]);

}

}

protected void findMarkets()

{

ddlMarket.Items.Clear();

foo.Market[0] = "--Choose--";

for (int w = 1; w < 2000; w++)

{

foo.Market[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT MARKET " +

"FROM OPERATOR\_INFORMATION WHERE Len(MARKET) > 0 " +

"ORDER BY MARKET ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.Market[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlMarket.Items.Add(foo.Market[c]);

}

}

protected void findSiteIDs()

{

ddlSiteID.Items.Clear();

foo.SiteID[0] = "--Choose--";

for (int w = 1; w < 20000; w++)

{

foo.SiteID[w] = "";

}

int i = 1;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DISTINCT SITE\_ID " +

"FROM SITE\_INFORMATION1 WHERE Len(SITE\_ID) > 0 " +

"ORDER BY SITE\_ID ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.SiteID[i] = reader[0].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

}

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

{

ddlSiteID.Items.Add(foo.SiteID[c]);

}

}

//Function to get operator information

private void GetOperatorInfo()

{

foo.ispopulated = 0;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT CARRIER, MARKET, RF\_TECHNOLOGY, VENDOR\_PO\_NUMBER FROM OPERATOR\_INFORMATION " +

"WHERE SITE\_ID = @SiteID";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.Add(new SqlParameter("@SiteID", //the name of the parameter to map

System.Data.SqlDbType.NVarChar, //SqlDbType value

20, //The width of the parameter

"Site\_ID")); //The name of the column source

//Fill the parameter with the value retrieved

//from the text field

cmd.Parameters["@SiteID"].Value = foo.Site\_ID;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.ispopulated = 1;

foo.checksite = "ttt";

generalInfo.Carrier = reader[0].ToString();

generalInfo.Market = reader[1].ToString();

generalInfo.RFTechnology = reader[2].ToString();

generalInfo.VendorPO = reader[3].ToString();

}

// Call Close when done reading.

reader.Close();

if (foo.ispopulated == 0)

{

generalInfo.Carrier = "";

generalInfo.Market = "";

generalInfo.RFTechnology = "";

generalInfo.VendorPO = "";

}

}

}

//Function to get site information

private void GetSiteInfo()

{

foo.ispopulated = 0;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT SITE\_NAME, LATITUDE, LONGITUDE, ADDRESS, CITY, STATE, SITE\_CONTACT, CONTACT\_EMAIL, CONTACT\_PHONE, " +

"ACCESS\_CODE, ACCESS\_INSTRUCTIONS, TEMP\_SITE\_ID, SITE\_ID FROM SITE\_INFORMATION1 " +

"WHERE SITE\_ID = @SiteID";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.Add(new SqlParameter("@SiteID", //the name of the parameter to map

System.Data.SqlDbType.NVarChar, //SqlDbType value

20, //The width of the parameter

"Site\_ID")); //The name of the column source

//Fill the parameter with the value retrieved

//from the text field

cmd.Parameters["@SiteID"].Value = foo.Site\_ID;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.ispopulated = 1;

generalInfo.SiteName = reader[0].ToString();

generalInfo.Latitude = reader[1].ToString();

generalInfo.Longitute = reader[2].ToString();

generalInfo.SiteAddress = reader[3].ToString();

generalInfo.City = reader[4].ToString();

generalInfo.State = reader[5].ToString();

generalInfo.SiteContact = reader[6].ToString();

generalInfo.ContactEmail = reader[7].ToString();

generalInfo.ContactPhone = reader[8].ToString();

generalInfo.AccessCode = reader[9].ToString();

generalInfo.AccessInstructions = reader[10].ToString();

generalInfo.TempSiteID = reader[11].ToString();

generalInfo.SiteID = reader[12].ToString();

}

// Call Close when done reading.

reader.Close();

if (foo.ispopulated == 0)

{

generalInfo.SiteName = "";

generalInfo.Latitude = "";

generalInfo.Longitute = "";

generalInfo.SiteAddress = "";

generalInfo.City = "";

generalInfo.State = "";

generalInfo.SiteContact = "";

generalInfo.ContactEmail = "";

generalInfo.ContactPhone = "";

generalInfo.AccessCode = "";

generalInfo.AccessInstructions = "";

generalInfo.SiteID = "";

generalInfo.TempSiteID = "";

}

}

}

protected void GetBTSInfo()

{

foo.ispopulated = 0;

int test = 0;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT VENDOR\_TYPE, MODEL, CABINET\_FRAMES, EXPANSION\_FRAMES " +

" FROM BTS\_INFORMATION " +

"WHERE SITE\_ID = @SiteID";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.Add(new SqlParameter("@SiteID", //the name of the parameter to map

System.Data.SqlDbType.NVarChar, //SqlDbType value

20, //The width of the parameter

"Site\_ID")); //The name of the column source

//Fill the parameter with the value retrieved

//from the text field

cmd.Parameters["@SiteID"].Value = foo.Site\_ID;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.ispopulated = 1;

test = 1;

generalInfo.VendorType = reader[0].ToString();

generalInfo.Model = reader[1].ToString();

generalInfo.cabinetFrames = reader[2].ToString();

generalInfo.ExpansionFrames = reader[3].ToString();

}

if (test == 1)

tabdone.generaldone = 1;

else

tabdone.generaldone = 0;

if (foo.ispopulated == 0)

{

generalInfo.VendorType = "Lucent";

generalInfo.Model = "";

generalInfo.cabinetFrames = "1";

generalInfo.ExpansionFrames = "1";

}

// Call Close when done reading.

reader.Close();

}

}

protected void GetAdditionalInfo()

{

foo.ispopulated = 0;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT PO\_NUMBER, NUMBER\_OF\_SECTORS " +

"FROM GENERAL\_ADDITIONAL " +

"WHERE SITE\_ID = @SiteID";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.Add(new SqlParameter("@SiteID", //the name of the parameter to map

System.Data.SqlDbType.NVarChar, //SqlDbType value

20, //The width of the parameter

"Site\_ID")); //The name of the column source

//Fill the parameter with the value retrieved

//from the text field

cmd.Parameters["@SiteID"].Value = foo.Site\_ID;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.ispopulated = 1;

generalInfo.CustomerSitePO = reader[0].ToString();

generalInfo.NumberOfSectors = reader[1].ToString();

}

// Call Close when done reading.

reader.Close();

if (foo.ispopulated == 0)

{

generalInfo.CustomerSitePO = "";

generalInfo.NumberOfSectors = "1";

}

}

}

//Pull frequency info from database if exists for site

protected void GetFrequencyInfo()

{

foo.ispopulated = 0;

int test = 0;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT FREQUENCY\_BAND\_OF\_OPERATION1, NUMBER\_OF\_ACTIVE\_CARRIERS1, " +

"FREQUENCY\_BAND\_OF\_OPERATION2, NUMBER\_OF\_ACTIVE\_CARRIERS2, " +

"FREQUENCY\_BAND\_OF\_OPERATION3, NUMBER\_OF\_ACTIVE\_CARRIERS3, " +

"FREQUENCY\_BAND\_OF\_OPERATION4, NUMBER\_OF\_ACTIVE\_CARRIERS4 " +

"FROM FREQUENCY\_BANDS " +

"WHERE SITE\_ID = @SiteID";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.Add(new SqlParameter("@SiteID", //the name of the parameter to map

System.Data.SqlDbType.NVarChar, //SqlDbType value

20, //The width of the parameter

"SiteID")); //The name of the column source

//Fill the parameter with the value retrieved

//from the text field

cmd.Parameters["@SiteID"].Value = foo.Site\_ID;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.ispopulated = 1;

test = 1;

rftableInfo.FrequencyBandOfOperation1 = reader[0].ToString();

rftableInfo.NumActiveCarriers1 = reader[1].ToString();

rftableInfo.FrequencyBandOfOperation2 = reader[2].ToString();

rftableInfo.NumActiveCarriers2 = reader[3].ToString();

rftableInfo.FrequencyBandOfOperation3 = reader[4].ToString();

rftableInfo.NumActiveCarriers3 = reader[5].ToString();

rftableInfo.FrequencyBandOfOperation4 = reader[6].ToString();

rftableInfo.NumActiveCarriers4 = reader[7].ToString();

}

if (test == 1)

tabdone.rfdone = 1;

else

tabdone.rfdone = 0;

// Call Close when done reading.

reader.Close();

if (foo.ispopulated == 0)

{

rftableInfo.FrequencyBandOfOperation1 = "A";

rftableInfo.NumActiveCarriers1 = "1";

rftableInfo.FrequencyBandOfOperation2 = "A";

rftableInfo.NumActiveCarriers2 = "1";

rftableInfo.FrequencyBandOfOperation3 = "A";

rftableInfo.NumActiveCarriers3 = "1";

rftableInfo.FrequencyBandOfOperation4 = "A";

rftableInfo.NumActiveCarriers4 = "1";

}

}

}

//Pull sector info from database if exists for that site

protected void GetSectorInfo()

{

foo.ispopulated = 0;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT AZIMUTH\_ALPHA, AZIMUTH\_BETA, AZIMUTH\_GAMMA, AZIMUTH\_ALPHA1, AZIMUTH\_BETA1, AZIMUTH\_GAMMA1, " +

"RFPC\_ALPHA, RFPC\_BETA, RFPC\_GAMMA, RFPC\_ALPHA1, RFPC\_BETA1, RFPC\_GAMMA1, " +

"DIPLEXED\_ALPHA, DIPLEXED\_BETA, DIPLEXED\_GAMMA, DIPLEXED\_ALPHA1, DIPLEXED\_BETA1, DIPLEXED\_GAMMA1, " +

"DUPLEXED\_ALPHA, DUPLEXED\_BETA, DUPLEXED\_GAMMA, DUPLEXED\_ALPHA1, DUPLEXED\_BETA1, DUPLEXED\_GAMMA1, " +

"TMA\_ALPHA, TMA\_BETA, TMA\_GAMMA, TMA\_ALPHA1, TMA\_BETA1, TMA\_GAMMA1 " +

"FROM NUMBER\_OF\_SECTORS " +

"WHERE SITE\_ID = @SiteID";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.Add(new SqlParameter("@SiteID", //the name of the parameter to map

System.Data.SqlDbType.NVarChar, //SqlDbType value

20, //The width of the parameter

"SiteID")); //The name of the column source

//Fill the parameter with the value retrieved

//from the text field

cmd.Parameters["@SiteID"].Value = foo.Site\_ID;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.ispopulated = 1;

rftableInfo.AzimuthAlpha = reader[0].ToString();

rftableInfo.AzimuthBeta = reader[1].ToString();

rftableInfo.AzimuthGamma = reader[2].ToString();

rftableInfo.AzimuthAlpha1 = reader[3].ToString();

rftableInfo.AzimuthBeta1 = reader[4].ToString();

rftableInfo.AzimuthGamma1 = reader[5].ToString();

rftableInfo.RFPCAlpha = reader[6].ToString();

rftableInfo.RFPCBeta = reader[7].ToString();

rftableInfo.RFPCGamma = reader[8].ToString();

rftableInfo.RFPCAlpha1 = reader[9].ToString();

rftableInfo.RFPCBeta1 = reader[10].ToString();

rftableInfo.RFPCGamma1 = reader[11].ToString();

rftableInfo.DiplexedAlpha = reader[12].ToString();

rftableInfo.DiplexedBeta = reader[13].ToString();

rftableInfo.DiplexedGamma = reader[14].ToString();

rftableInfo.DiplexedAlpha1 = reader[15].ToString();

rftableInfo.DiplexedBeta1 = reader[16].ToString();

rftableInfo.DiplexedGamma1 = reader[17].ToString();

rftableInfo.DuplexedAlpha = reader[18].ToString();

rftableInfo.DuplexedBeta = reader[19].ToString();

rftableInfo.DuplexedGamma = reader[20].ToString();

rftableInfo.DuplexedAlpha1 = reader[21].ToString();

rftableInfo.DuplexedBeta1 = reader[22].ToString();

rftableInfo.DuplexedGamma1 = reader[23].ToString();

rftableInfo.TMAAlpha = reader[24].ToString();

rftableInfo.TMABeta = reader[25].ToString();

rftableInfo.TMAGamma = reader[26].ToString();

rftableInfo.TMAAlpha1 = reader[27].ToString();

rftableInfo.TMABeta1 = reader[28].ToString();

rftableInfo.TMAGamma1 = reader[29].ToString();

}

// Call Close when done reading.

reader.Close();

if (foo.ispopulated == 0)

{

rftableInfo.AzimuthAlpha = "0";

rftableInfo.AzimuthBeta = "0";

rftableInfo.AzimuthGamma = "0";

rftableInfo.AzimuthAlpha1 = "0";

rftableInfo.AzimuthBeta1 = "0";

rftableInfo.AzimuthGamma1 = "0";

rftableInfo.RFPCAlpha = "1";

rftableInfo.RFPCBeta = "1";

rftableInfo.RFPCGamma = "1";

rftableInfo.RFPCAlpha1 = "1";

rftableInfo.RFPCBeta1 = "1";

rftableInfo.RFPCGamma1 = "1";

rftableInfo.DiplexedAlpha = "Yes";

rftableInfo.DiplexedBeta = "Yes";

rftableInfo.DiplexedGamma = "Yes";

rftableInfo.DiplexedAlpha1 = "Yes";

rftableInfo.DiplexedBeta1 = "Yes";

rftableInfo.DiplexedGamma1 = "Yes";

rftableInfo.DuplexedAlpha = "Yes";

rftableInfo.DuplexedBeta = "Yes";

rftableInfo.DuplexedGamma = "Yes";

rftableInfo.DuplexedAlpha1 = "Yes";

rftableInfo.DuplexedBeta1 = "Yes";

rftableInfo.DuplexedGamma1 = "Yes";

rftableInfo.TMAAlpha = "TMA";

rftableInfo.TMABeta = "TMA";

rftableInfo.TMAGamma = "TMA";

rftableInfo.TMAAlpha1 = "TMA";

rftableInfo.TMABeta1 = "TMA";

rftableInfo.TMAGamma1 = "TMA";

}

}

}

protected void GetAncillaryInfo()

{

foo.ispopulated = 0;

int test = 0;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT ANTENNAE\_TYPE, CIRCUIT\_BREAKER\_TYPE, POWER\_SUPPLY, CB\_FUSE\_SIZE, RACK\_SIZE\_WALL\_MOUNT, " +

"HVAC\_MODEL, BATTERY\_PLANT\_MODEL\_TYPE, BTS\_T1\_COUNT " +

" FROM PHYSICAL\_ATTRIBUTES " +

"WHERE SITE\_ID = @SiteID";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.Add(new SqlParameter("@SiteID", //the name of the parameter to map

System.Data.SqlDbType.NVarChar, //SqlDbType value

20, //The width of the parameter

"Site\_ID")); //The name of the column source

//Fill the parameter with the value retrieved

//from the text field

cmd.Parameters["@SiteID"].Value = foo.Site\_ID;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.ispopulated = 1;

tabdone.ancillarydone = 1;

ancillary.AntennaeType = reader[0].ToString();

ancillary.CircuitBreakerType = reader[1].ToString();

ancillary.PowerSupply = reader[2].ToString();

ancillary.CBFuseSize = reader[3].ToString();

ancillary.RackSizeWallMount = reader[4].ToString();

ancillary.HVACModel = reader[5].ToString();

ancillary.BatteryPlantModelType = reader[6].ToString();

ancillary.BTST1Count = reader[7].ToString();

}

if (test == 1)

tabdone.ancillarydone = 1;

else

tabdone.ancillarydone = 0;

// Call Close when done reading.

reader.Close();

if (foo.ispopulated == 0)

{

ancillary.AntennaeType = "DAS";

ancillary.CircuitBreakerType = "";

ancillary.PowerSupply = "";

ancillary.CBFuseSize = "";

ancillary.RackSizeWallMount = "19 inches";

ancillary.HVACModel = "";

ancillary.BatteryPlantModelType = "";

ancillary.BTST1Count = "Yes";

}

}

}

protected void GetSiteNotes()

{

foo.ispopulated = 0;

notes.AuditorName = "";

notes.Notes = "";

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT DIAGRAM\_TITLE, DIAGRAM\_DESCRIPTION " +

"FROM SITE\_NOTES " +

"WHERE SITE\_ID = @SiteID";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.Add(new SqlParameter("@SiteID", //the name of the parameter to map

System.Data.SqlDbType.NVarChar, //SqlDbType value

20, //The width of the parameter

"Site\_ID")); //The name of the column source

//Fill the parameter with the value retrieved

//from the text field

cmd.Parameters["@SiteID"].Value = foo.Site\_ID;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

foo.ispopulated = 1;

notes.AuditorName = reader[0].ToString();

notes.Notes = reader[1].ToString();

}

// Call Close when done reading.

reader.Close();

if (foo.ispopulated == 0)

{

notes.AuditorName = "";

notes.Notes = "";

}

}

}

protected void GetFloorPlans()

{

int test = 0;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT \* FROM SITE\_DIAGRAMS " +

"WHERE SITE\_ID = @SiteID";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.Add(new SqlParameter("@SiteID", //the name of the parameter to map

System.Data.SqlDbType.NVarChar, //SqlDbType value

20, //The width of the parameter

"Site\_ID")); //The name of the column source

//Fill the parameter with the value retrieved

//from the text field

cmd.Parameters["@SiteID"].Value = foo.Site\_ID;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

test = 1;

}

if (test == 1)

tabdone.floorplansdone = 1;

else

tabdone.floorplansdone = 0;

// Call Close when done reading.

reader.Close();

}

}

protected void GetPhotos()

{

int test = 0;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT \* FROM SITE\_PHOTOS " +

"WHERE SITE\_ID = @SiteID";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.Add(new SqlParameter("@SiteID", //the name of the parameter to map

System.Data.SqlDbType.NVarChar, //SqlDbType value

20, //The width of the parameter

"Site\_ID")); //The name of the column source

//Fill the parameter with the value retrieved

//from the text field

cmd.Parameters["@SiteID"].Value = foo.Site\_ID;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

test = 1;

}

if (test == 1)

tabdone.photosdone = 1;

else

tabdone.photosdone = 0;

// Call Close when done reading.

reader.Close();

}

}

protected void getGeneralTimeStamps()

{

int i = 0;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT \* FROM GENERAL\_TIMESTAMP " +

"WHERE SITE\_ID = @SiteID ORDER BY UNIT\_NUMBER ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.Add(new SqlParameter("@SiteID", //the name of the parameter to map

System.Data.SqlDbType.NVarChar, //SqlDbType value

20, //The width of the parameter

"Site\_ID")); //The name of the column source

//Fill the parameter with the value retrieved

//from the text field

cmd.Parameters["@SiteID"].Value = foo.Site\_ID;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

generalInfo.generaltimestamp[i] = reader[1].ToString();

generalInfo.usernamegeneral[i] = reader[2].ToString();

generalInfo.unitnumbergeneral[i] = reader[3].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

generalInfo.generalTime = i;

}

}

protected void getRFTimeStamps()

{

int i = 0;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT \* FROM RF\_TIMESTAMP " +

"WHERE SITE\_ID = @SiteID ORDER BY UNIT\_NUMBER ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.Add(new SqlParameter("@SiteID", //the name of the parameter to map

System.Data.SqlDbType.NVarChar, //SqlDbType value

20, //The width of the parameter

"Site\_ID")); //The name of the column source

//Fill the parameter with the value retrieved

//from the text field

cmd.Parameters["@SiteID"].Value = foo.Site\_ID;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

rftableInfo.rftimestamp[i] = reader[1].ToString();

rftableInfo.usernamerf[i] = reader[2].ToString();

rftableInfo.unitnumberrf[i] = reader[3].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

rftableInfo.rfTime = i;

}

}

protected void getAncillaryTimeStamps()

{

int i = 0;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT \* FROM ANCILLARY\_TIMESTAMP " +

"WHERE SITE\_ID = @SiteID ORDER BY UNIT\_NUMBER ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.Add(new SqlParameter("@SiteID", //the name of the parameter to map

System.Data.SqlDbType.NVarChar, //SqlDbType value

20, //The width of the parameter

"Site\_ID")); //The name of the column source

//Fill the parameter with the value retrieved

//from the text field

cmd.Parameters["@SiteID"].Value = foo.Site\_ID;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

ancillary.ancillarytimestamp[i] = reader[1].ToString();

ancillary.usernameancillary[i] = reader[2].ToString();

ancillary.unitnumberancillary[i] = reader[3].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

ancillary.ancillaryTime = i;

}

}

protected void getNotesTimeStamps()

{

int i = 0;

SqlCommand cmd = null;

string connectionString = ConfigurationManager.ConnectionStrings["PSCV1ConnectionString"].ConnectionString;

string queryString = @"SELECT \* FROM NOTES\_TIMESTAMP " +

"WHERE SITE\_ID = @SiteID ORDER BY UNIT\_NUMBER ASC";

using (SqlConnection connection =

new SqlConnection(connectionString))

{

SqlCommand command =

new SqlCommand(queryString, connection);

connection.Open();

cmd = new SqlCommand(queryString);

cmd.Connection = connection;

cmd.Parameters.Add(new SqlParameter("@SiteID", //the name of the parameter to map

System.Data.SqlDbType.NVarChar, //SqlDbType value

20, //The width of the parameter

"Site\_ID")); //The name of the column source

//Fill the parameter with the value retrieved

//from the text field

cmd.Parameters["@SiteID"].Value = foo.Site\_ID;

SqlDataReader reader = cmd.ExecuteReader();

// Call Read before accessing data.

while (reader.Read())

{

notes.notestimestamp[i] = reader[1].ToString();

notes.usernamenotes[i] = reader[2].ToString();

notes.unitnumbernotes[i] = reader[3].ToString();

i++;

}

// Call Close when done reading.

reader.Close();

notes.notesTime = i;

}

}

protected void btnLogOut\_Click(object sender, EventArgs e)

{

login.UserName = "";

login.role = "";

Session["login"] = login;

Response.Redirect("~/Login.aspx");

}

protected void Page\_PreRender(object sender, EventArgs e)

{

Session["Table\_Info1"] = generalInfo;

Session["Table\_Info2"] = rftableInfo;

Session["Table\_Info3"] = ancillary;

Session["tabdone1"] = tabdone;

Session["notes"] = notes;

Session["login.aspx"] = login;

}

protected void btnDownload\_Click(object sender, ImageClickEventArgs e)

{

}

protected void btnDownload\_Click(object sender, EventArgs e)

{

}

}