ARTICLE

READER LEVEL:

How to Bind GridView and perform edit update and delete operation on gridview in 3 Tiers in ASP.Net Using C#

By Nilesh Jadav on May 25, 2015

In this article you will learn how to Bind GridView in 3 tier in ASP.Net using C# with Stored Procedure.

In this article I will show you how to Bind GridView in 3 tier in ASP.Net using C# with Stored Procedure. We will also edit, delete and update the data in GridView.

Initial Chamber

Step 1: Open Visual Studio 2010 and create an Empty Website. Give it a suitable name [GridView_demo].

Step 2: In Solution Explorer you will get your empty website. Add a web form, SQL database and 3 class files. By going like this:

For Web Form

GridView_demo (Your Empty Website) -> Right Click -> Add New Item -> Web Form. Name it as -> gridview_demo.aspx.

For SQL Server Database

GridView_demo (Your Empty Website) -> Right Click -> Add New Item -> SQL Server Database. [Add Database inside the App_Data_folder].

For 3 Class Files

GridView_demo (Your Empty Website) -> Right Click -> Add New Item -> Class [Add 3 Class files - -> Add your class file in App_code Folder] - -> Give name as the following:

1. Commonfunctions.cs
2. BAL_user_operation.cs

3. DAL_user_operation.cs

Database Chamber

Step 3: In Server Explorer, Click on your Database [Database.mdf] - -> Tables - -> Add New Table -:- Make table like the following:

Table - -> tbl_data [Don't Forget to make ID as IS Identity -- True]

dbo.tbl_data: Ta...ATA\DATABASE.MDF) | dbo.sp_update: S...TA\DA

	Column Name	Data Type	Allow Nulls
🔑	id	int	☐
	name	varchar(50)	☑
	city	varchar(50)	☑
▶	email	varchar(50)	☑
			☐

Add some Stored procedure for Update, Insert and Delete Data by going to Database [Database.mdf] - -> Stored Procedures - -> Right Click - -> Add New Stored Procedures.

1. sp_getdata()

dbo.sp_getdata:...ATA\DATABASE.MDF) | dbo

```
ALTER PROCEDURE sp_getdata
  /*
  (
  @parameter1 int = 5,
  @parameter2 datatype OUTPUT
  )
  */
AS
  select * from tbl_data
  RETURN
```

2. sp_insert()

dbo.sp_insert: St...ATA\DATABASE.MDF) | dbo.sp_getdata:...A

```
ALTER PROCEDURE sp_insert
(
  @name varchar(50),
  @city varchar(50),
  @email varchar(50)


)
AS
  insert into tbl_data values(@name,@city,@email)
  RETURN
```

3. sp_update()

dbo.sp_update: S...TA\DATABASE.MDF)

```
ALTER PROCEDURE sp_update
  (
    @id int,
    @name varchar(50),
    @city varchar(50),
    @email varchar(50)
  )
AS
  update tbl_data
  set
  name= @name,
  city=@city,
  email=@email
  where id=@id
  RETURN
```

4. sp_delete()

dbo.sp_delete: S...ATA\DATABASE.MDF)

```
ALTER PROCEDURE sp_delete
(

  @id int

)
AS
  delete from tbl_data where id = @id
  RETURN
```

These all are the Stored Procedures that we will use for updating, deleting and editing data in GridView.

Design Code

Step 5: Now it's time for serious design in GridView. Let's begin by opening gridview_demo.aspx page and try the following code:

```
<body>
    <form id="form1" runat="server">

    <table style="width:100%;">
        <caption class="style3">
            <strong>Bind Grid View Using 3Tier</strong></caption>
        <tr>

            <td  align="center">

    <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
        BackColor="LightGoldenrodYellow" BorderColor="Tan" BorderWidth="1px"
        CellPadding="2" DataKeyNames="id" ForeColor="Black" GridLines="None"
                    AutoGenerateDeleteButton="True" AutoGenerateEditButton="True"
                    onrowcancelingedit="GridView1_RowCancelingEdit"
                    onrowdeleting="GridView1_RowDeleting" onrowediting="GridView1_Row
                    onrowupdating="GridView1_RowUpdating">
        <AlternatingRowStyle BackColor="PaleGoldenrod" />
```

```
        <Columns>
            <asp:TemplateField HeaderText="Name" SortExpression="name">
                <EditItemTemplate>
                    <asp:TextBox ID="TextBox1" runat="server" Text='<%# Bind("name")
</asp:TextBox>
                </EditItemTemplate>
                <ItemTemplate>
                    <asp:Label ID="Label1" runat="server" Text='<%# Bind("name") %>'>
</asp:Label>
```

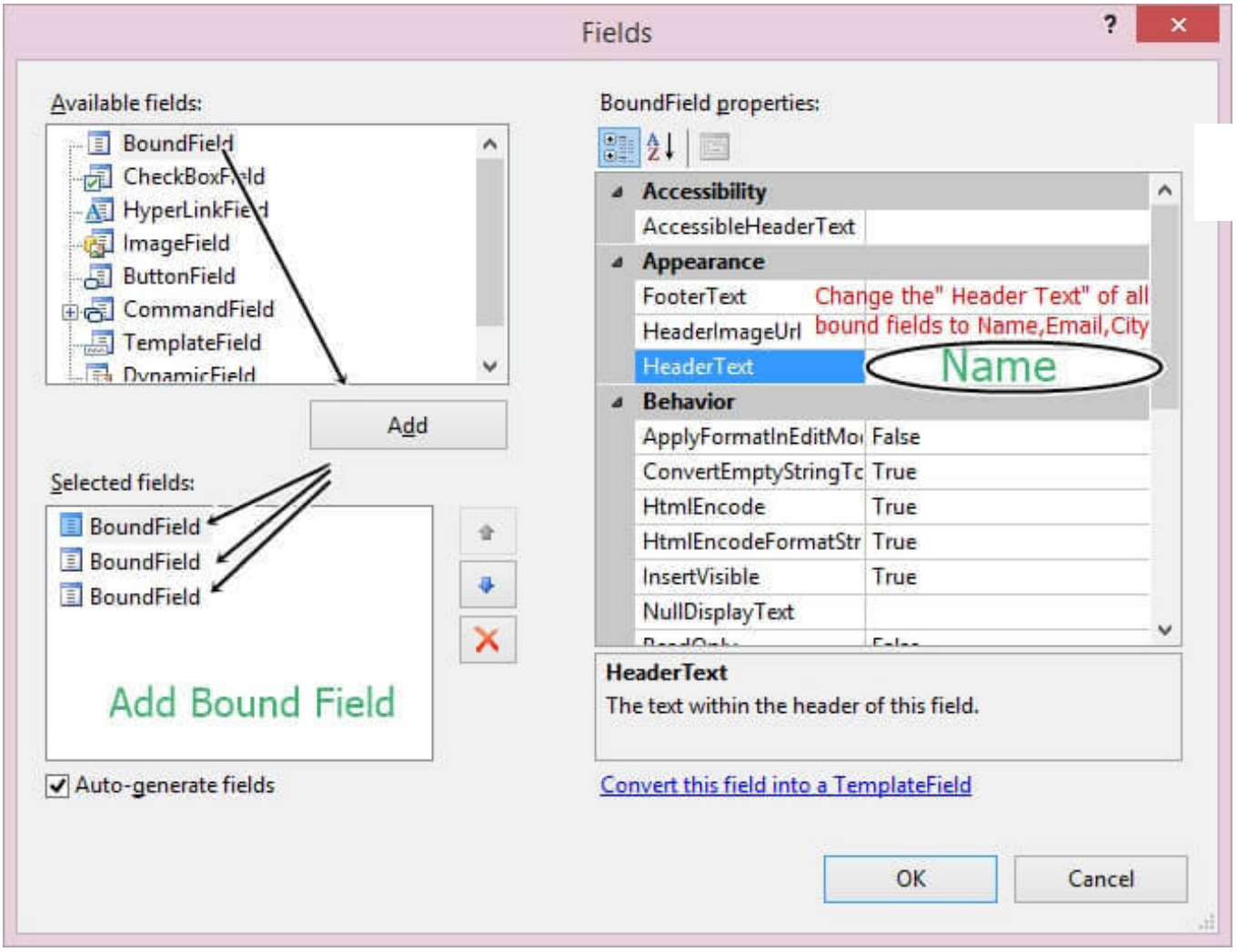
 ID="Label3" runat="server" Text='<%# Bind("email") %>'

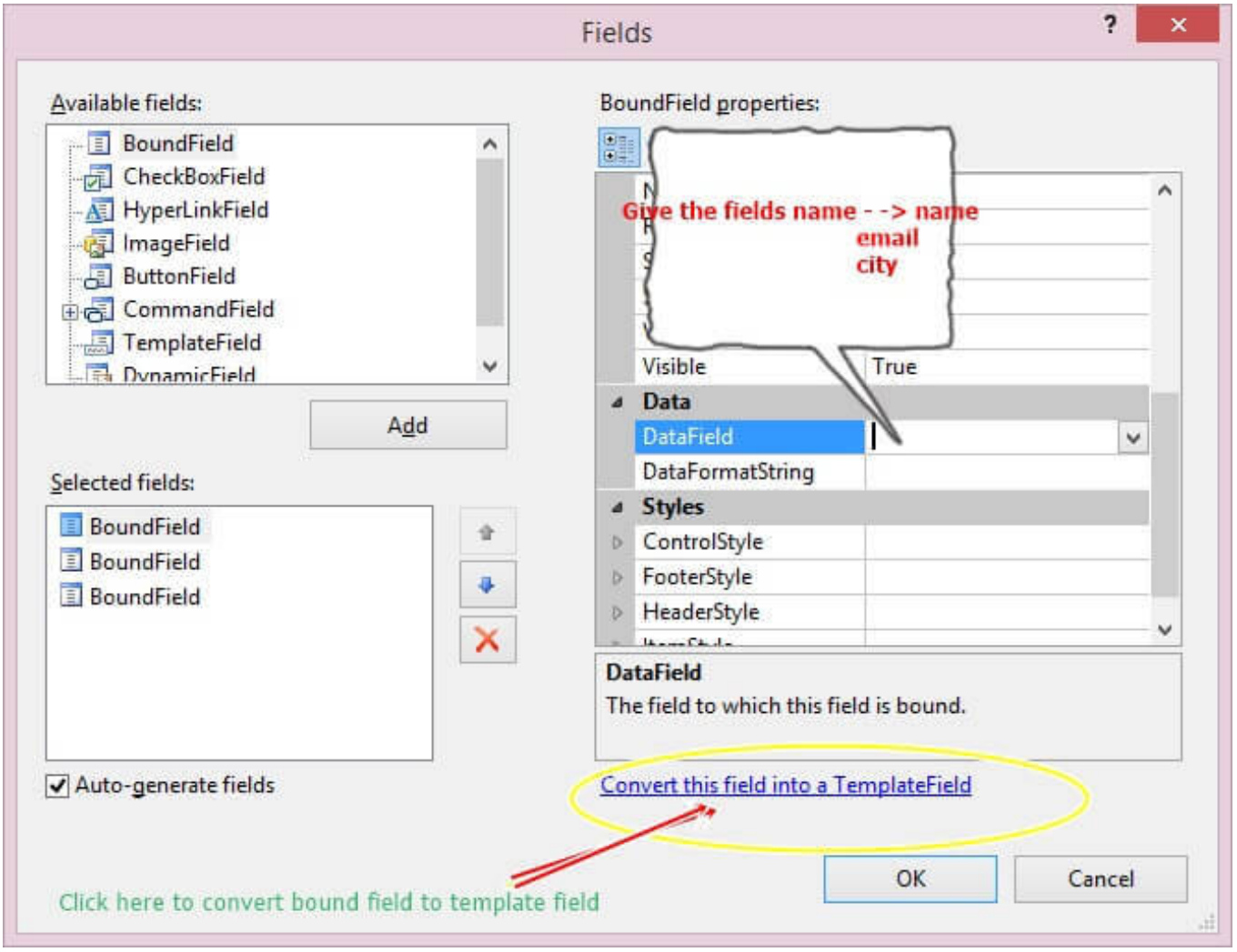
```
                </ItemTemplate>
            </asp:TemplateField>
            <asp:TemplateField HeaderText="City" SortExpression="city">
                <EditItemTemplate>
                    <asp:TextBox ID="TextBox2" runat="server" Text='<%# Bind("city")
</asp:TextBox>
                </EditItemTemplate>
                <ItemTemplate>
                    <asp:Label ID="Label2" runat="server" Text='<%# Bind("city") %>'>
</asp:Label>
                </ItemTemplate>
            </asp:TemplateField>
        </Columns>
        <FooterStyle BackColor="Tan" />
        <HeaderStyle BackColor="Tan" Font-Bold="True" />
        <PagerStyle BackColor="PaleGoldenrod" ForeColor="DarkSlateBlue"
            HorizontalAlign="Center" />
        <SelectedRowStyle BackColor="DarkSlateBlue" ForeColor="GhostWhite" />
        <SortedAscendingCellStyle BackColor="#FAFAE7" />
        <SortedAscendingHeaderStyle BackColor="#DAC09E" />
        <SortedDescendingCellStyle BackColor="#E1DB9C" />
        <SortedDescendingHeaderStyle BackColor="#C2A47B" />
    </asp:GridView>
            </td>

        </tr>
    </table>
    </form>
</body>
```

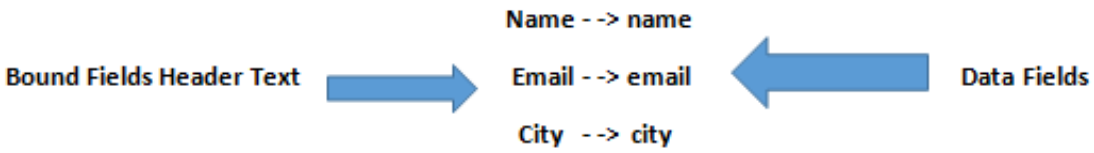
You can also manually create this design by dragging the GridView into .aspx page. Then click the arrow sign on GridView - -> GridView tasks will open - -> Edit Columns - -> A “Field” window will open (following image). Here you have to add three “BoundField” Button from - -> Available Fields. - -> Change the header text - -> Name, Email, City.

Unclick the Auto Generated Button at the bottom [Note: I forgot to unclick it.].


Fields
Available fields:
BoundField
CheckBoxField
HyperLinkField
ImageField
ButtonField
CommandField
TemplateField
DynamicField
Add
Selected fields:
BoundField
BoundField
BoundField
Add Bound Field
Auto-generate fields
BoundField properties:
Accessibility
AccessibleHeaderText
Appearance
FooterText
HeaderImageUrl
HeaderText
Name
Change the" Header Text" of all bound fields to Name,Email,City
Behavior
ApplyFormatInEditMo
ConvertEmptyStringTo
HtmlEncode
HtmlEncodeFormatStr
InsertVisible
NullDisplayText
False
True
True
True
True
HeaderText
The text within the header of this field.
Convert this field into a TemplateField
OK
Cancel



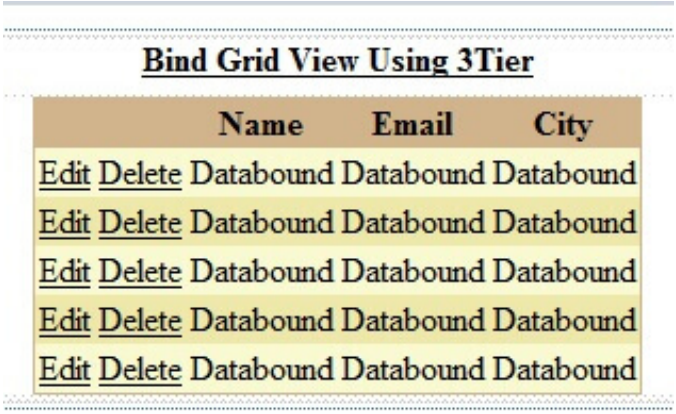
Get into Bound Field Properties - -> Data Fields - -> Change every Bound field's Data field to - ->



Go to GridView [In design mode] - -> press F4 to open Property window of GridView and find - -> Data Keys Name - -> and write – id.

In Property Window find - -> Auto Generate Edit and Auto Generate Delete Button and make it - -> True.

This is your actual Design.



Code Chamber

1. Open Commonfunction.cs file and add this code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Configuration;

/// <summary>
/// Summary description for Commonfunctions
/// </summary>
public class Commonfunctions
{
    public Commonfunctions()
    {
        //
        // TODO: Add constructor logic here
        //
    }
    public static string getconstring()
    {

        return ConfigurationManager.ConnectionStrings["dbcon"].ToString();

    }
}
```

The preceding code is written for SQL Connection String that we have to call again and again by going to the database property. This is a lengthy process, that's why we made this class and now we will just call its method - ->getconstring() that will make our process shorter and comfortable.

2. Open DAL_user_operation.cs file and code it as in the following code.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data;
using System.Data.SqlClient;

/// <summary>
/// Summary description for DAL_user_operation
/// </summary>
public class DAL_user_operation
{
    public DAL_user_operation()
```

```
14.     {
15.         //
16.         // TODO: Add constructor logic here
17.         //
18.     }
19.     public bool user_insert(string name, string email,string city)
20.     {
21.         SqlConnection con = new SqlConnection(Commonfunctions.getconstring());
22.         SqlCommand cmd = new SqlCommand("sp_insert", con);
23.         cmd.CommandType = CommandType.StoredProcedure;
24.         cmd.Parameters.AddWithValue("name", name);
25.         cmd.Parameters.AddWithValue("email",email);
26.         cmd.Parameters.AddWithValue("city", city);
27.
28.         con.Open();
29.         int i = cmd.ExecuteNonQuery();
30.        con.Close();
31.         if (i != 0)
32.         {
33.             return true;
34.         }
35.         else
36.         {
37.
38.             return false;
39.
40.         }
41.
42.     }
43.     public void  user_delete(int id)
44.     {
45.
46.         SqlConnection con = new SqlConnection(Commonfunctions.getconstring());
47.         SqlCommand cmd = new SqlCommand("sp_delete", con);
48.         cmd.CommandType = CommandType.StoredProcedure;
49.         cmd.Parameters.AddWithValue("id", id);
50.         con.Open();
51.         int i = cmd.ExecuteNonQuery();
52.         con.Close();
53.
54.     }
55.
56.     public void user_update(string name, string email,string city, int id)
57.     {
58.
59.         SqlConnection con = new SqlConnection(Commonfunctions.getconstring());
60.         SqlCommand cmd = new SqlCommand("sp_update", con);
61.         cmd.CommandType = CommandType.StoredProcedure;
62.         cmd.Parameters.AddWithValue("name", name);
63.         cmd.Parameters.AddWithValue("email",email);
64.         cmd.Parameters.AddWithValue("city", city);
65.
66.         cmd.Parameters.AddWithValue("id", id);
67.         con.Open();
68.         int i = cmd.ExecuteNonQuery();
69.         con.Close();
70.
71.
72.     }
73.     public DataTable getdata()
74.     {
75.
76.         SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachD
77.         SqlCommand cmd = new SqlCommand("sp_getdata", con);
78.         cmd.CommandType = CommandType.StoredProcedure;
79.         SqlDataAdapter sda = new SqlDataAdapter(cmd);
80.         DataTable dt = new DataTable();
81.       sda.Fill(dt);
82.
83.         return dt;
```

```
    }
}
```

3. Open your BAL_user_operation.cs File and code it like the following:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data;
using System.Data.SqlClient;

/// <summary>
/// Summary description for BAL_user_operation
/// </summary>
public class BAL_user_operation
{
    DAL_user_operation du = new DAL_user_operation();
    public BAL_user_operation()
    {
        //
        // TODO: Add constructor logic here
        //
    }

    public bool user_insert(string name, string email,string city)
    {
        return du.user_insert(name, email,city);
    }

    public void  user_delete(int id)
    {
         du.user_delete(id);

    }

    public void user_update(string name, string email,string city, int id)
    {

        du.user_update(name, email, city, id);
    }

    public DataTable getdata()
   {
       return du.getdata();
    }
}
```

4. At last Open gridview_demo.aspx.cs file and code it like the following.

```
System;
System.Collections.Generic;
System.Linq;
System.Web;
System.Web.UI;
System.Web.UI.WebControls;

: partial class _Default : System.Web.UI.Page

AL_user_operation bu = new BAL_user_operation();
otected void Page_Load(object sender, EventArgs e)

   if (!Page.IsPostBack)
   {
       refreshdata();
```

```
    }

ublic void refreshdata()


    GridView1.DataSource = bu.getdata();
    GridView1.DataBind();


otected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)

    int id = Convert.ToInt16(GridView1.DataKeys[e.RowIndex].Values["id"].ToString());
    bu.user_delete(id);
    refreshdata();


otected void GridView1_RowEditing(object sender, GridViewEditEventArgs e)

    GridView1.EditIndex = e.NewEditIndex;
    refreshdata();


otected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs e)


    TextBox txtname = GridView1.Rows[e.RowIndex].FindControl("TextBox1") as TextBox;
    TextBox txtemail = GridView1.Rows[e.RowIndex].FindControl("TextBox3") as TextBox;
    TextBox txtcity = GridView1.Rows[e.RowIndex].FindControl("TextBox2") as TextBox;

  int id= Convert.ToInt16(GridView1.DataKeys[e.RowIndex].Values["id"].ToString());
  bu.user_update(txtname.Text, txtemail.Text, txtcity.Text, id);
  GridView1.EditIndex = -1;
  refreshdata();

otected void GridView1_RowCancelingEdit(object sender, GridViewCancelEditEventArgs e)

    GridView1.EditIndex = -1;
    refreshdata();
```

web.config file code

```
<configuration>

    <system.web>
        <compilation debug="true" targetFramework="4.0" />
    </system.web>
  <connectionStrings>

    <add name="dbcon" connectionString="Data Source=.\SQLEXPRESS;AttachDbFilename=|Da

  </connectionStrings>

</configuration>
```

Output Chamber

Bind Grid View Using 3Tier

	Name	Email	City
Edit Delete	Nilesh Jadav	nilusilu3@gmail.com	Rajkot
Edit Delete	Purnima	nilushilu@yahoo.com	Rajkot
Edit Delete	Rinku	nilusilu4@rediff.com	Ahmedabad
Edit Delete	Chandni	nilusilu3@hotmail.com	Mumbai

Bind Grid View Using 3Tier

	Name	Email	City
Update Cancel	Nilesh Jadav	nilusilu3@gmail.com	Rajkot
Edit Delete	Purnima	nilushilu@yahoo.com	Rajkot
Edit Delete	Rinku	nilusilu4@rediff.com	Ahmedabad
Edit Delete	Chandni	nilusilu3@hotmail.com	Mumbai

Hope you liked this!
Have a nice day and enjoy this tutorial.

Download 100% FREE Spire Office APIs

asp.net Bind c# delete Edit GridView how Jadav Nilesh procedure store Tier update

NILESH JADAV TOP 100

Nilesh is professionally an ASP.NET developer, SEO and Blog writer at Citadel Infotech. He is having competent knowledge of ASP.NET, C#, HTML, AJAX, ADO.NET, Linq, SQL Server, JavaScript, PHP, Wordpress and other CMS. I... Read more

http://nileshjadav.blogspot.in/

62 Rank | 1.5m Read | Platinum Member | 1 Time

COMMENTS (19)

TRENDING UP

01 Using Generics With C#

02 Web Crawling With C#

03 What Is The Future Of XAML

04 Best Practices For ASP.NET MVC Application

05 Best Programming Language Of 2016

06 Introduction To .NET Core 1.0

07 Factory Design Pattern Real World Example

08 Highest Paying Tech Jobs Of 2016

09 How To Become A Microsoft MVP

10 Keep Learning, Or Become Dust!

View All