Day-14

Database creation

```
Use [GridData]

CREATE TABLE [dbo].[OperatingSystem](
[OSId] [int] IDENTITY(1,1) NOT NULL,
[OSName] [varchar](100) NULL,
[CreateDate] [datetime] NULL,
[Status] [smallint] NULL,
CONSTRAINT [PK_OperatingSystem] PRIMARY KEY
CLUSTERED
(
[OSId] ASC
)WITH (PAD_INDEX = OFF,
STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
```

```
SET ANSI_PADDING OFF
GO

ALTER TABLE [dbo].[OperatingSystem] ADD DEFAULT
(getdate()) FOR [CreateDate]
GO

ALTER TABLE [dbo].[OperatingSystem] ADD DEFAULT ('1')
FOR [Status]
GO



//Create Procedure
Create Procedure [dbo].[InsertSystemName]
(
@OSName varchar(50)
)
AS
BEGIN
insert into OperatingSystem(OSName) values(@OSName)
END

//Get Data Procedure
ALTER Procedure [dbo].[GetSystemsData]
As
Begin
select * from OperatingSystem where Status='1'
End

ALTER Procedure [dbo].[UpdateSystems]
(
@Id int,
@Name varchar(100),
@Status int
)
As
BEGIN
update OperatingSystem set OSName=@Name,Status=@Status
where OSId=@Id
END

//Delete Procedure
ALTER Procedure [dbo].[DeleteSystemsData]
(
@Id int
)
As
Begin
update OperatingSystem set Status='0' where OSId=@Id

End
```

Databaselayer-SqlHelper.cs

```
public static string CONNECTION_STRING =
ConfigurationManager.ConnectionStrings["Data
Source=WIN-VU1R6I95I3U;Initial
Catalog=AssignmentDb;Integrated
Security=True"].ConnectionString;
```

```
public static DataSet
ExecuteParamerizedSelectCommand(string
CommandName, CommandType cmdType,
SqlParameter[] param)
{
   DataSet ds = new DataSet();

   using (SqlConnection con = new
SqlConnection(CONNECTION_STRING))
   {
      using (SqlCommand cmd = con.CreateCommand())
      {
         cmd.CommandType = cmdType;
         cmd.CommandText = CommandName;
         cmd.Parameters.AddRange(param);
         try
         {
            if (con.State != ConnectionState.Open)
            {
               con.Open();
            }

            using (SqlDataAdapter da = new
SqlDataAdapter(cmd))
            {
               da.Fill(ds);
            }
         }
```

```
            catch
            {
                throw;
            }
        }
    }
    return ds;
}

public static bool ExecuteNonQuery(string
CommandName, CommandType cmdType,
SqlParameter[] pars)
{
    int result = 0;

    using (SqlConnection con = new
SqlConnection(CONNECTION_STRING))
    {
        using (SqlCommand cmd = con.CreateCommand())
        {
            cmd.CommandType = cmdType;
            cmd.CommandText = CommandName;
            cmd.Parameters.AddRange(pars);

            try
            {
                if (con.State != ConnectionState.Open)
                {
                    con.Open();
                }
                result = cmd.ExecuteNonQuery();
            }
            catch
            {
                throw;
            }
        }
    }
    return (result > 0);
}
```

EntityLayer-GridEntity.CS

```
public class OperatingSystemEntity
{
    public int OSId
    {
        get;
        set;
    }
    public string OSName
    {
        get;
        set;
    }
    public DateTime CreateDate
    {
        get;
        set;
    }
    public int Status
    {
        get;
        set;
    }
}
```

DAL-GridData.cs

```
public class GridData
{
    public bool CreateSystem(OperatingSystemEntity
SEntity)
    {
        SqlParameter[] parameters = new SqlParameter[] {
            new SqlParameter("@OSName",
SEntity.OSName),

        };
        return
SqlHelper.ExecuteNonQuery("InsertSystemName",
CommandType.StoredProcedure, parameters);
    }

    public List<OperatingSystemEntity>
GetSystemsData(OperatingSystemEntity SEntity)
    {
        List<OperatingSystemEntity> ListEntry = null;
        SqlParameter[] parameters = new SqlParameter[] {

        };
        using (DataSet ds =
SqlHelper.ExecuteParamerizedSelectCommand("GetSyst
emsData", CommandType.StoredProcedure, parameters))
        {
            if (ds.Tables.Count > 0)
            {

                ListEntry = new
List<OperatingSystemEntity>();

                foreach (DataRow row2 in ds.Tables[0].Rows)
                {
                    OperatingSystemEntity entry = new
OperatingSystemEntity();
                    entry.OSId =
Convert.ToInt32(row2["OSId"].ToString());
```

```
                    entry.OSName =
row2["OSName"].ToString();
                    entry.CreateDate =
Convert.ToDateTime(row2["CreateDate"].ToString());
                    entry.Status =
Convert.ToInt32(row2["Status"].ToString());
                    ListEntry.Add(entry);
                }
            }
        }

        return ListEntry;
    }

    public bool UpdateSystems(OperatingSystemEntity
SEntity)
    {
        SqlParameter[] parameters = new SqlParameter[] {
            new SqlParameter("@Id", SEntity.OSId),
                new SqlParameter("@Name",
SEntity.OSName),
                new SqlParameter("@Status", SEntity.Status),
        };

        return
SqlHelper.ExecuteNonQuery("UpdateSystems",
CommandType.StoredProcedure, parameters);
    }
    public bool DeleteSystem(OperatingSystemEntity
SEntity)
    {
        SqlParameter[] parameters = new SqlParameter[] {
            new SqlParameter("@Id", SEntity.OSId),
        };

        return
SqlHelper.ExecuteNonQuery("DeleteSystemsData",
CommandType.StoredProcedure, parameters);
    }
}
}
```

NewGrid.aspx

```
< asp:Content ID = "Content1"
ContentPlaceHolderID="head" runat="server">
     <script language="javascript" type="text/javascript">
     function validate() {

    if (document.getElementById("<%=txtname.ClientID
%>").value == "")
    {
        alert("Please Enter Name");
        document.getElementById("<%=txtname.ClientID
%>").focus();
        return false;
    }

}
     </ script >
</ asp:Content >
 < asp:Content ID = "Content2"
ContentPlaceHolderID="Body" runat="server">
     <div>
          <h2 style="width: 80%">
               <u>Create System</u></h2>
          <table>
               <tr>
                    <td colspan="2">
                         <asp:Label runat = "server"
ID="lblmsgerror" Font-Bold="true"></asp:Label >

                      </ td >

                 </ tr >

                 < tr >

                      < td >
```

```
                   < asp:Label ID = "lblname"
runat="server" Text="SystemName"></asp:Label >

                  </ td >

                  < td >

                     < asp:TextBox ID = "txtname"
runat="server"></asp:TextBox >

                    </ td >

                 </ tr >

                 < tr >

                    < td >

                    </ td >

                    < td >

                       < asp:Button ID = "btnsave"
runat="server" Text="Save" OnClick="btnsave_Click" />
               </td>
            </tr>
         </table>
      </div>
```

```
     <h2 style="width: 80%">
          <u>Total Availble Systems</u></h2>
     <p>
          <asp:Label runat = "server" ID="lblmsg"
ForeColor="Red" Font-Bold="true"></asp:Label >

      </ p >

      < div >

           < asp:GridView ID = "gvSystem" runat="server"
AutoGenerateColumns="false"
OnRowUpdating="gvSystem_RowUpdating"
OnRowEditing="gvSystem_RowEditing"
OnRowDeleting="gvSystem_RowDeleting"
DataKeyNames="OSId"
OnRowCancelingEdit="gvSystem_RowCancelingEdit">
               <Columns>
                    <asp:TemplateField >
                         < EditItemTemplate >
                              < asp:ImageButton ID =
"imgbtnUpdate" CommandName="Update"
runat="server" ImageUrl="~/Images/Update.jpg"
ToolTip="Update" Height="20px" Width="20px" />
                              <asp:ImageButton ID =
"imgbtnCancel" runat="server"
CommandName="Cancel"
```

```
ImageUrl="~/Images/Cancel.jpg" ToolTip="Cancel"
Height="20px" Width="20px" />
                    </EditItemTemplate>
                    <ItemTemplate>
                        <asp:ImageButton ID = "imgbtnEdit"
CommandName="Edit" runat="server"
ImageUrl="~/Images/Edit.jpg" ToolTip="Edit"
Height="20px" Width="20px" />
                        <asp:ImageButton ID =
"imgbtnDelete" CommandName="Delete"
OnClientClick="return confirm('Are you sure you want to
delete selected record?')" Text="Edit" runat="server"
ImageUrl="~/images/Delete.jpg" ToolTip="Delete"
Height="20px" Width="20px" />
                    </ItemTemplate>
                </asp:TemplateField >
                < asp:TemplateField HeaderText = "System
Name" >

                     < EditItemTemplate >

                         < asp:TextBox ID = "txtname"
runat="server" Text='<%#
Eval("OSName")%>'></asp:TextBox >

                       </ EditItemTemplate >

                       < ItemTemplate >
```

```
                              < asp:Label ID = "lblname"
runat="server" Text='<%#
Eval("OSName")%>'></asp:Label >

                            </ ItemTemplate >

                        </ asp:TemplateField >

                        < asp:TemplateField HeaderText =
"Date Of Created" >

                              < ItemTemplate >

                                   < asp:Label ID = "lbldate"
runat="server" Text='<%#
Eval("CreateDate")%>'></asp:Label >

                                 </ ItemTemplate >

                            </ asp:TemplateField >

                             < asp:TemplateField HeaderText =
"Status" >

                                   < EditItemTemplate >
                                   < asp:TextBox ID =
"txtstatus" runat="server" Text='<%#
Eval("Status")%>'></asp:TextBox >

                                 </ EditItemTemplate >

                                 < ItemTemplate >

                                    < asp:Label ID = "lblstatus"
runat="server" Text='<%#
Eval("Status")%>'></asp:Label >

                                   </ ItemTemplate >

                                </ asp:TemplateField >

                            </ Columns >

                         </ asp:GridView >

                      </ div >
                   </ asp:Content >
```

NewGrid.aspx.cs

```
OperatingSystemEntity SEntity = new
OperatingSystemEntity();
```

```
GridData GData = new GridData();

protected void Page_Load(object sender, EventArgs e)
{
    btnsave.Attributes.Add("onclick", "return validate()");
    if (!IsPostBack)
    {
        BindData();
    }
}

protected void btnsave_Click(object sender, EventArgs e)
{
    SEntity.OSName = txtname.Text;

    if (GData.CreateSystem(SEntity) == true)
    {
        lblmsgerror.ForeColor = Color.Green;
        lblmsgerror.Text = "System Name Saved
Successfully";
        BindData();
        txtname.Text = "";
    }
    else
    {
        lblmsgerror.ForeColor = Color.Red;
        lblmsgerror.Text = "System Name Already Exists ";
    }
```

```
}

private void BindData()
{
    List<OperatingSystemEntity> SList =
GData.GetSystemsData(SEntity);

    if (SList.Count != 0)
    {
        gvSystem.DataSource = SList;
        gvSystem.DataBind();
    }
    else
    {
        lblmsg.Text = "No Data found..";
    }
}

private static String GetTextFromRowBox(GridViewRow
row, String field)
{
    return ((TextBox)row.FindControl(field)).Text;
}

protected void gvSystem_RowUpdating(object sender,
GridViewUpdateEventArgs e)
{
    int id =
Convert.ToInt32(gvSystem.DataKeys[e.RowIndex].Value
s["OSId"].ToString());
    GridViewRow row = gvSystem.Rows[e.RowIndex];

    SEntity.OSName = GetTextFromRowBox(row,
"txtname");
    SEntity.Status =
Convert.ToInt32(GetTextFromRowBox(row,
"txtstatus").ToString());
    SEntity.OSId = id;

    if (GData.UpdateSystems(SEntity) == true)
    {
        gvSystem.EditIndex = -1;
        BindData();
        lblmsg.Text = "Records Updated sucessfully";
    }
    else
    {
        lblmsg.Text = "Updation Failed";
    }
}

protected void gvSystem_RowEditing(object sender,
GridViewEditEventArgs e)
{
    gvSystem.EditIndex = e.NewEditIndex;
    BindData();
}

protected void gvSystem_RowCancelingEdit(object
sender, GridViewCancelEditEventArgs e)
{
    gvSystem.EditIndex = -1;
    BindData();
}

protected void gvSystem_RowDeleting(object sender,
GridViewDeleteEventArgs e)
{
    int ID =
Convert.ToInt32(gvSystem.DataKeys[e.RowIndex].Value
s["OSId"].ToString());

    deleterecords(ID);
}

private void deleterecords(int ID)
{
    SEntity.OSId = ID;

    if (GData.DeleteSystem(SEntity) == true)
    {
        lblmsg.Text = "Records deleted sucessfully...";
        BindData();
        lblmsg.Text = "";
    }
    else
    {
        lblmsg.ForeColor = Color.Red;
        lblmsg.Text = "Records deleted failed...";
        BindData();
        lblmsg.Text = "";
    }
}
```