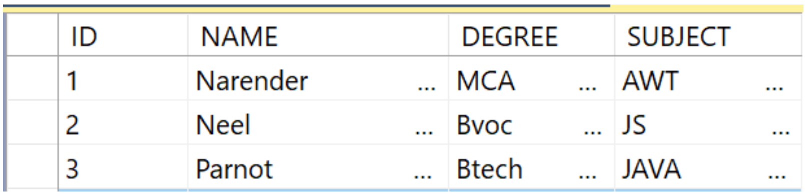
AIM: DEMONSTRATE USAGE OF DATA BOUND CONTROLS, SIMPLE STORED PROCEDURE AND PARAMETERIZED STORED PROCEDURES IN ASP.NET

WebConfig:

```
<add name="ApplicationServices" connectionString="server=. ; database=narender_p4;
Trusted_Connection=Yes;" providerName="System.Data.SqlClient"/>
```

DATABASE:



	ID	NAME	DEGREE	SUBJECT
	1	Narender ...	MCA ...	AWT ...
	2	Neel ...	Bvoc ...	JS ...
	3	Parnot ...	Btech ...	JAVA ...

A) Create a web page that demonstrates the use of data bound controls of ASP.NET.

SOURCE CODE:

DataBound.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="DataBound.aspx.cs"
Inherits="DataBound" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
   <h2>Display Data using Connected Architecture</h2>
   <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
CellPadding="6" OnRowCancelingEdit="GridView1_RowCancelingEdit"
OnRowEditing="GridView1_RowEditing" OnRowUpdating="GridView1_RowUpdating">
         <Columns>
            <asp:TemplateField>
               <ItemTemplate>
                  <asp:Button ID="btn_Edit" runat="server" Text="Edit" CommandName="Edit"
/>
               </ItemTemplate>
               <EditItemTemplate>
                  <asp:Button ID="btn_Update" runat="server" Text="Update"
CommandName="Update"/>
                  <asp:Button ID="btn_Cancel" runat="server" Text="Cancel"
CommandName="Cancel"/>
               </EditItemTemplate>
```

```
        </asp:TemplateField>
        <asp:TemplateField HeaderText="ID">
          <ItemTemplate>
            <asp:Label ID="lbl_ID" runat="server" Text='<%#Eval("ID") %>'></asp:Label>
          </ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="NAME">
          <ItemTemplate>
            <asp:Label ID="lbl_Name" runat="server" Text='<%#Eval("NAME")
%>'></asp:Label>
          </ItemTemplate>
          <EditItemTemplate>
            <asp:TextBox ID="txt_Name" runat="server" Text='<%#Eval("NAME")
%>'></asp:TextBox>
          </EditItemTemplate>
        </asp:TemplateField>
         <asp:TemplateField HeaderText="COURSE">
          <ItemTemplate>
            <asp:Label ID="lbl_Course" runat="server" Text='<%#Eval("DEGREE")
%>'></asp:Label>
          </ItemTemplate>
          <EditItemTemplate>
            <asp:TextBox ID="txt_Course" runat="server" Text='<%#Eval("DEGREE")
%>'></asp:TextBox>
          </EditItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="SUBJECT">
          <ItemTemplate>
            <asp:Label ID="lbl_Subject" runat="server" Text='<%#Eval("SUBJECT")
%>'></asp:Label>
          </ItemTemplate>
          <EditItemTemplate>
            <asp:TextBox ID="txt_Subject" runat="server" Text='<%#Eval("SUBJECT")
%>'></asp:TextBox>
          </EditItemTemplate>
        </asp:TemplateField>
      </Columns>
      <HeaderStyle BackColor="#663300" ForeColor="#ffffff"/>
      <RowStyle BackColor="#e7ceb6"/>
    </asp:GridView>
  </div>
  </form>
</body>
</html>
```

DataBound.aspx.cs:

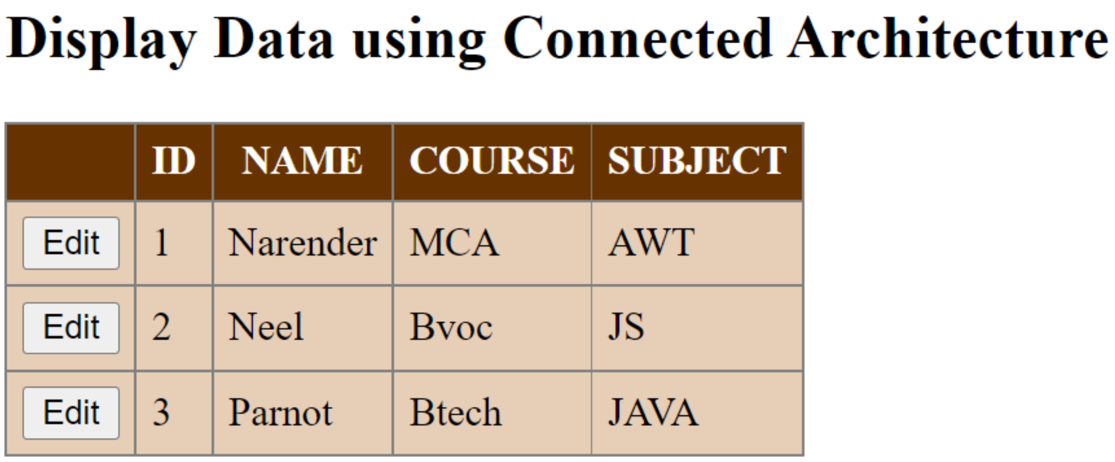
```
using System;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
```

```
public partial class DataBound : System.Web.UI.Page
{
   //Connection String from web.config File
   string cs =
ConfigurationManager.ConnectionStrings["ApplicationServices"].ConnectionString;
   SqlConnection con;
   SqlDataAdapter adapt;
   DataTable dt;
   protected void Page_Load(object sender, EventArgs e)
   {
      if(!IsPostBack)
      {
         ShowData();
      }
   }
   //ShowData method for Displaying Data in Gridview
   protected void ShowData()
   {
      dt = new DataTable();
      con = new SqlConnection(cs);
      con.Open();
      adapt = new SqlDataAdapter("Select ID,NAME,DEGREE,SUBJECT from student",con);
      adapt.Fill(dt);
      if(dt.Rows.Count>0)
      {
         GridView1.DataSource = dt;
         GridView1.DataBind();
      }
      con.Close();
   }

   protected void GridView1_RowEditing(object sender,
System.Web.UI.WebControls.GridViewEditEventArgs e)
   {
      //NewEditIndex property used to determine the index of the row being edited.
      GridView1.EditIndex = e.NewEditIndex;
      ShowData();
   }
   protected void GridView1_RowUpdating(object sender,
System.Web.UI.WebControls.GridViewUpdateEventArgs e)
   {
      //Finding the controls from Gridview for the row which is going to update
      Label id=GridView1.Rows[e.RowIndex].FindControl("lbl_ID") as Label;
      TextBox name = GridView1.Rows[e.RowIndex].FindControl("txt_NAME") as TextBox;
      TextBox course = GridView1.Rows[e.RowIndex].FindControl("txt_COURSE") as TextBox;
      TextBox subject = GridView1.Rows[e.RowIndex].FindControl("txt_SUBJECT") as TextBox;
      con = new SqlConnection(cs);
      con.Open();
      //updating the record
```

```
        SqlCommand cmd = new SqlCommand("Update student set
NAME='"+name.Text+"',DEGREE='"+course.Text+"',SUBJECT='"+subject.Text+"' where
ID="+Convert.ToInt32(id.Text),con);
        cmd.ExecuteNonQuery();
        con.Close();
        //Setting the EditIndex property to -1 to cancel the Edit mode in Gridview
        GridView1.EditIndex = -1;
        //Call ShowData method for displaying updated data
        ShowData();
    }
    protected void GridView1_RowCancelingEdit(object sender,
System.Web.UI.WebControls.GridViewCancelEditEventArgs e)
    {
        //Setting the EditIndex property to -1 to cancel the Edit mode in Gridview
        GridView1.EditIndex = -1;
        ShowData();
    }
}
```

OUTPUT:

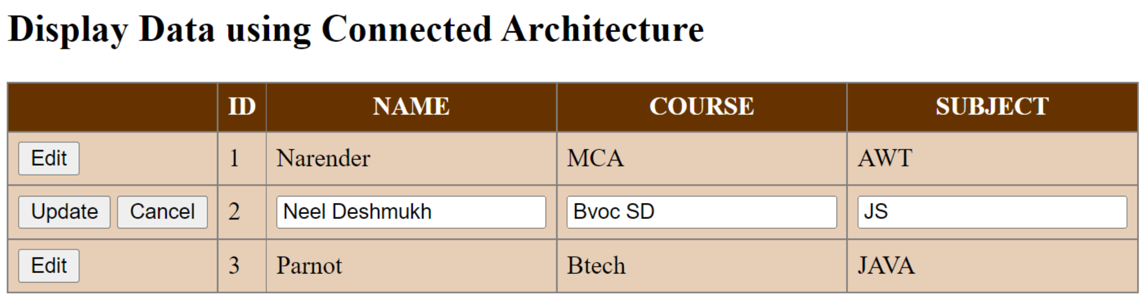
BEFORE EDITING:



Display Data using Connected Architecture

	ID	NAME	COURSE	SUBJECT
Edit	1	Narender	MCA	AWT
Edit	2	Neel	Bvoc	JS
Edit	3	Parnot	Btech	JAVA

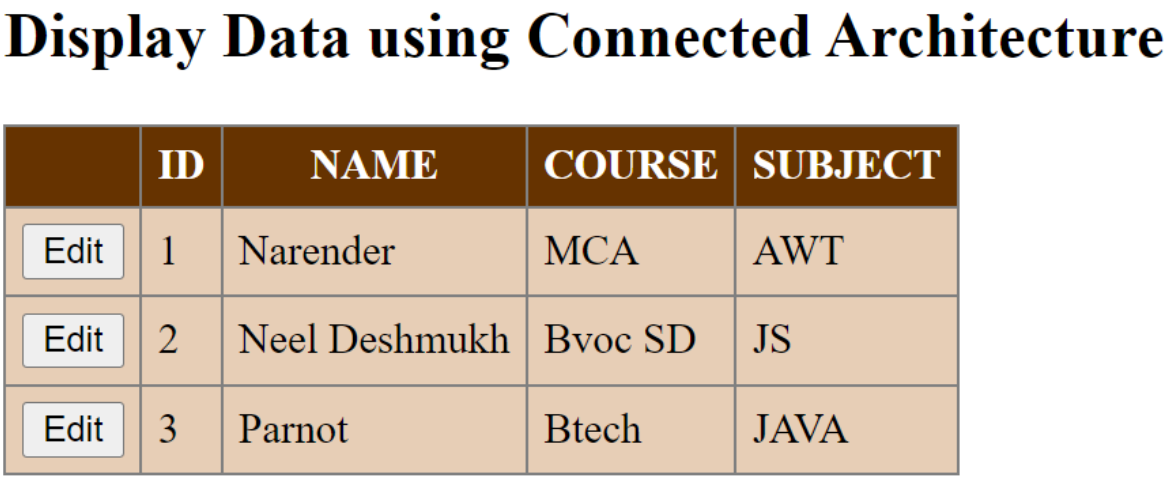
EDITING:



Display Data using Connected Architecture

	ID	NAME	COURSE	SUBJECT
Edit	1	Narender	MCA	AWT
Update Cancel	2	Neel Deshmukh	Bvoc SD	JS
Edit	3	Parnot	Btech	JAVA

AFTER EDITING:



Display Data using Connected Architecture

	ID	NAME	COURSE	SUBJECT
Edit	1	Narender	MCA	AWT
Edit	2	Neel Deshmukh	Bvoc SD	JS
Edit	3	Parnot	Btech	JAVA

B) **Design a web page to demonstrate the working of a simple stored procedure.**

DATABASE:

```
CREATE OR ALTER PROCEDURE simpleStudentProcedureByNarender
AS
select * from narender_p4.dbo.student
GO;
```

SOURCE CODE:

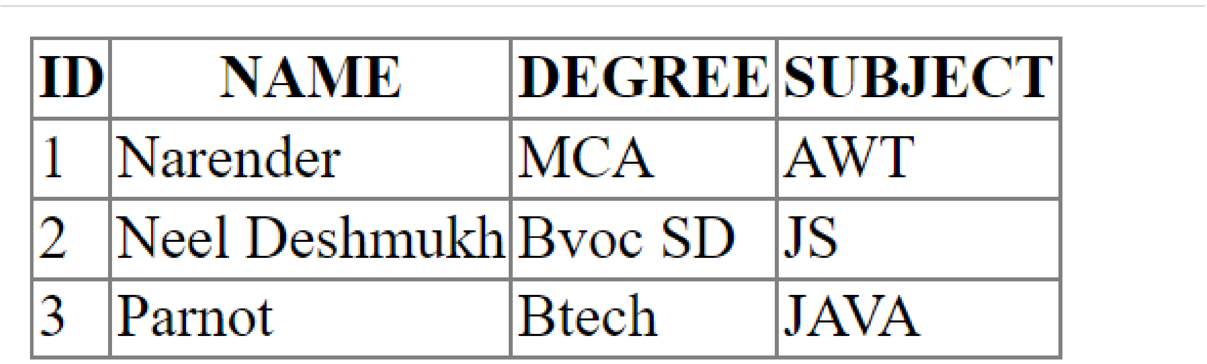
StoredProcedure.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="StoredProcedure.aspx.cs"
Inherits="StoredProcedure" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
    <asp:GridView ID="GridView1" runat="server">
     </asp:GridView>
  </div>
  </form>
</body>
</html>
```

StoredProcedure.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class StoredProcedure : System.Web.UI.Page
{
   string cs =
ConfigurationManager.ConnectionStrings["ApplicationServices"].ConnectionString;
   SqlConnection con;
   protected void Page_Load(object sender, EventArgs e)
   {
      con = new SqlConnection(cs);
      con.Open();

      System.Data.SqlClient.SqlCommand objCmd = new
System.Data.SqlClient.SqlCommand("simpleStudentProcedureByNarender", con);

   objCmd.CommandType = System.Data.CommandType.StoredProcedure;

   GridView1.DataSource = objCmd.ExecuteReader();

   GridView1.DataBind();

   con.Close();
   }
}
```

OUTPUT:



ID	NAME	DEGREE	SUBJECT
1	Narender	MCA	AWT
2	Neel Deshmukh	Bvoc SD	JS
3	Parnot	Btech	JAVA

C) Design a web page to demonstrate the working of parameterized stored procedure.

DATABASE:

```
CREATE OR ALTER PROCEDURE paramStudentProcedureByNarender(
@ID INT, @NAME NVARCHAR(50), @DEGREE NVARCHAR(50), @SUBJECT NVARCHAR(50) )
AS
BEGIN
INSERT INTO student( [ID], [NAME], [DEGREE], [SUBJECT] )
VALUES ( @ID, @NAME, @DEGREE, @SUBJECT )
END
```

SOURCE CODE:

ParameterSP.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ParameterSP.aspx.cs"
Inherits="ParameterSP" %>


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">


<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

   <title></title>

</head>

<body>

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

   <div>

            <asp:Label ID="Label1" runat="server" Text="ID"></asp:Label>

            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox><br /><br />

            <asp:Label ID="Label2" runat="server" Text="NAME"></asp:Label>

            <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox><br /><br />

            <asp:Label ID="Label3" runat="server" Text="DEGREE"></asp:Label>

            <asp:TextBox ID="TextBox3" runat="server"></asp:TextBox><br /><br />

            <asp:Label ID="Label4" runat="server" Text="SUBJECT"></asp:Label>

            <asp:TextBox ID="TextBox4" runat="server"></asp:TextBox><br /><br /><br />

            <asp:Button ID="Button1" runat="server" Text="Submit Record"
OnClick="Button1_Click" />
```

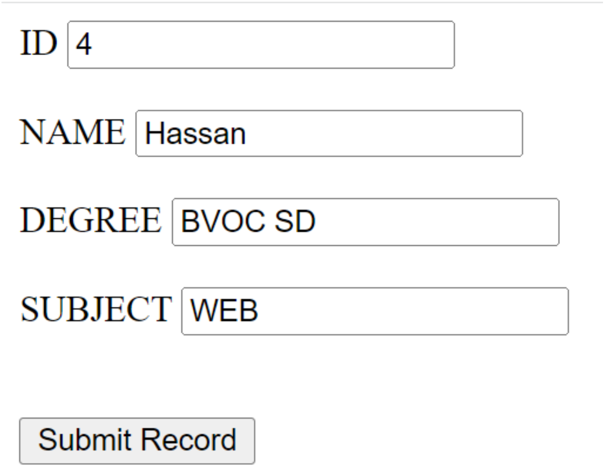
```
            <asp:Label ID="lblResult" runat="server" Text=""></asp:Label>

    </div>

    </form>

</body>

</html>
```

ParameterSP.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class ParameterSP : System.Web.UI.Page
{
    string cs = 
ConfigurationManager.ConnectionStrings["ApplicationServices"].ConnectionString;
    SqlConnection con;
    SqlCommand cmd = new SqlCommand();
    SqlParameter sp1, sp2, sp3, sp4;
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        con = new SqlConnection(cs);
        cmd = new SqlCommand("paramStudentProcedureByNarender", con);
        cmd.CommandType = CommandType.StoredProcedure;
        sp1 = new SqlParameter("@ID", SqlDbType.Int);
        sp2 = new SqlParameter("@NAME", SqlDbType.NVarChar);
        sp3 = new SqlParameter("@DEGREE", SqlDbType.NVarChar);
        sp4 = new SqlParameter("@SUBJECT", SqlDbType.NVarChar);
        sp1.Value = Convert.ToInt32(TextBox1.Text);
        sp2.Value = TextBox2.Text;
        sp3.Value = TextBox3.Text;
        sp4.Value = TextBox4.Text;
        cmd.Parameters.Add(sp1);
        cmd.Parameters.Add(sp2);
        cmd.Parameters.Add(sp3);
        cmd.Parameters.Add(sp4);
        con.Open();
        try
        {
          cmd.ExecuteNonQuery();
          lblResult.Text = "Successfully inserted!";
        }
        catch (Exception)
        {
          lblResult.Text = "Error was occured!";
        }
        con.Close();
    }
}
```

OUTPUT:

Before Inserting:



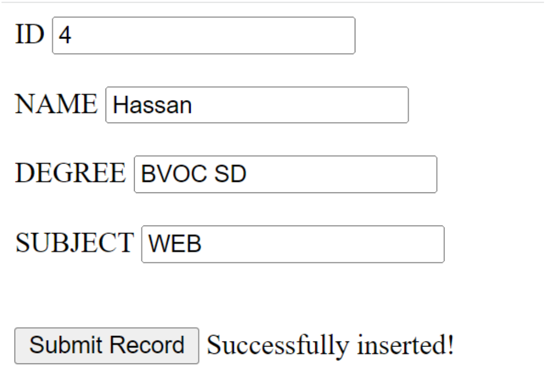
ID 4

NAME Hassan

DEGREE BVOC SD

SUBJECT WEB

After Inserting:



ID 4

NAME Hassan

DEGREE BVOC SD

SUBJECT WEB

Successfully inserted!

CONCLUSION:

From this practical, I have learned about data bound controls, simple stored procedure and parameterized stored procedures in ASP.NET.