| **Name of Student:** Ajay Karthikesan | | | |
| --- | --- | --- | --- |
| **Roll Number:** 57 | | **Assignment Number:** 6 | |
| **Aim of Assignment:**  Designing webpage to demonstrate use of data bound  controls, working of simple stored procedure and parameterized  stored procedure. | | | |
| **DOP:** 22.5.23 | | **DOS:** 29.5.23 | |
| **CO Mapped:**  CO2 | **PO Mapped:**  PO3, PO5, PSO1, PSO2 | **Faculty Signature:** | **Marks:** |

## 

## Practical No. 6

**Aim:** Designing webpage to demonstrate use of data bound

controls, working of simple stored procedure and parameterized

stored procedure.

Q.1) Create a webpage that demonstrates the use of data

bound controls of ASP.NET.

Q.2) Design a webpage to demonstrate the working of a

simple stored procedure.

Q.3) Design a webpage to demonstrate the working of

parameterized stored procedure.

**Theory:**

**Data Bound Controls:**

Databound controls are used to display data to the end-user

within the web applications and using databound controls allows

you to manipulate the data within the web applications very

easily.

Databound controls are bound to the DataSource property.

Databound controls are composite controls that combine other

ASP.NET Controls like Text boxes, Radio buttons, Buttons and

so on.

Frequently used Databound controls:

• Repeater

• List View

• DataList

• GridView

• Form View

**Repeater**

• Repeater controls is a Databound control to just display data in

the web application, using this we cannot manipulate the data; in

other words, a Repeater is a read-only control.

• Repeater is very light-weight and faster to display data

compared with other controls, so whenever you just want to

display a repeated list of items then use

**Repeater Control.**

• Repeater control works by repeating using the data source.

• Repeater Control appearance is controlled by its templates.

ListView:

The ListView control displays columns and rows of data and

allows sorting and paging. It is by far the most popular data

display control, and is ideal for understanding how data display

controls interact with data retrieval controls and code.

**DataList:**

DataList is a Databound control to display and manipulate data

in a web application. It is a composite control that can combine

other ASP.Net controls and it is present in the form. The

DataList appearance is controlled by its template fields.

The following template fields are supported by the DataList

control:

• Itemtemplate: It specifies the Items present in the Datasource,

it renders itself in the browser as many rows present in the data

source collection.

• EditItemTemplate: Used to provide edit permissions to the

user.

• HeaderTemplate: Used to display header text to the data source

collection.

• FooterTemplate: Used to display footer text to the data source

collection.

• ItemStyle: Used to apply styles to an ItemTemplate.

• EditStyle: Used to apply styles to an EditItemTemplate

• HeaderStyle: Used to apply styles to a HeaderTemplate

• FooterStyle: Used to apply styles to a FooterTemplate.

**GridView:**

The GridView control displays the values of a data source in a

table. Each column represents a field, while each row represents

a record. The GridView control supports the following features:

• Binding to data source controls, such as SqlDataSource.

• Built-in sort capabilities.

• Built-in update and delete capabilities.

• Built-in paging capabilities.

• Built-in row selection capabilities.

• Programmatic access to the GridView object model to

dynamically set properties, handle events, and so on.

• Multiple key fields.

• Multiple data fields for the hyperlink columns.

• Customizable appearance through themes and styles.

**FormView:**

The FormView control is used to display a single record at a

time from a data source. When you use the FormView control,

you create templates to display and edit data-bound values. The

templates contain controls, binding expressions, and formatting

that define the look and functionality of the form.

Stored procedures:

Stored procedures (sprocs) are generally an ordered series of

Transact-SQL statements bundled into a single logical unit.

They allow for variables and parameters, as well as selection

and looping constructs. A key point is that sprocs are stored in

the database rather than in a separate file.

Advantages over simply sending individual statements to the

server include:

• Referred to using short names rather than a long string of text;

therefore, less network traffic is required to run the code within

the sproc.

• Pre-optimized and precompiled, so they save an incremental

amount of time with each sproc call/execution.

• Encapsulate a process for added security or to simply hide the

complexity of the database.

• Can be called from other sprocs, making them reusable and

reducing code size.

**Parameterization Stored Procedure:**

A stored procedure gives us some procedural capability, and

also gives us a performance boost by using mainly two types of

parameters:

• Input parameters

• Output parameters

From outside the sproc, parameters can be passed in either by

position or reference.

Declaring Parameters in SQL Server Stored Procedures

• The name

• The datatype

• The default value

• The direction

**The syntax is:**

@parameter\_name [AS] datatype [= default|NULL]

[VARYING][OUTPUT|OUT]

**Code:**

File: StudentsService.cs

﻿using System;

using System.Data;

using System.Data.SqlClient;

namespace Practical6

{

public class StudentService

{

readonly SqlCommand cmd = new SqlCommand();

readonly SqlParameter sqlpId = new SqlParameter("@id", SqlDbType.Int);

readonly SqlParameter sqlpName = new SqlParameter("@Name", SqlDbType.VarChar);

readonly SqlParameter sqlpScore = new SqlParameter("@score", SqlDbType.Int);

internal StudentService()

{

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(sqlpId);

cmd.Parameters.Add(sqlpName);

cmd.Parameters.Add(sqlpScore);

cmd.CommandText = "ajayk57\_sql2";

}

internal bool AddStudentEntry(string id, string name, string score)

{

using (SqlConnection conn = new SqlConnection("Data Source=DESKTOP-OUL5TLP\\SQLEXPRESS;Initial Catalog=coll;Integrated Security=True"))

{

cmd.Connection = conn;

sqlpId.Value = id;

sqlpName.Value = name;

sqlpScore.Value = score;

conn.Open();

try

{

cmd.ExecuteNonQuery();

return true;

}

catch (Exception exc)

{

Console.WriteLine(exc.Message);

return false;

}

}

}

}

}

File: WebForm1.aspx

﻿<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="Practical6.WebForm1" %>

<!DOCTYPE html>

<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" AutoGenerateColumns="False" DataKeyNames="id" DataSourceID="SqlDataSource1">

<Columns>

<asp:BoundField DataField="id" HeaderText="id" ReadOnly="True" SortExpression="id" />

<asp:BoundField DataField="name" HeaderText="name" SortExpression="name" />

<asp:BoundField DataField="score" HeaderText="score" SortExpression="score" />

</Columns>

</asp:GridView>

<br />

<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:collConnectionString %>" OnSelecting="SqlDataSource1\_Selecting" ProviderName="<%$ ConnectionStrings:collConnectionString.ProviderName %>" SelectCommand="SELECT \* FROM [ajayk57\_student]"></asp:SqlDataSource>

<asp:FormView ID="FormView1" runat="server" BorderStyle="Dotted" DataKeyNames="id" DataSourceID="SqlDataSource1" OnPageIndexChanging="FormView1\_PageIndexChanging">

<EditItemTemplate>

id:

<asp:Label ID="idLabel1" runat="server" Text='<%# Eval("id") %>' />

<br />

name:

<asp:TextBox ID="nameTextBox" runat="server" Text='<%# Bind("name") %>' />

<br />

score:

<asp:TextBox ID="scoreTextBox" runat="server" Text='<%# Bind("score") %>' />

<br />

<asp:LinkButton ID="UpdateButton" runat="server" CausesValidation="True" CommandName="Update" Text="Update" />

&nbsp;<asp:LinkButton ID="UpdateCancelButton" runat="server" CausesValidation="False" CommandName="Cancel" Text="Cancel" />

</EditItemTemplate>

<InsertItemTemplate>

id:

<asp:TextBox ID="idTextBox" runat="server" Text='<%# Bind("id") %>' />

<br />

name:

<asp:TextBox ID="nameTextBox" runat="server" Text='<%# Bind("name") %>' />

<br />

score:

<asp:TextBox ID="scoreTextBox" runat="server" Text='<%# Bind("score") %>' />

<br />

<asp:LinkButton ID="InsertButton" runat="server" CausesValidation="True" CommandName="Insert" Text="Insert" />

&nbsp;<asp:LinkButton ID="InsertCancelButton" runat="server" CausesValidation="False" CommandName="Cancel" Text="Cancel" />

</InsertItemTemplate>

<ItemTemplate>

id:

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

<br />

name:

<asp:Label ID="nameLabel" runat="server" Text='<%# Bind("name") %>' />

<br />

score:

<asp:Label ID="scoreLabel" runat="server" Text='<%# Bind("score") %>' />

<br />

</ItemTemplate>

</asp:FormView>

<br />

<asp:DataList ID="DataList1" runat="server" BorderStyle="Groove" DataKeyField="id" DataSourceID="SqlDataSource1" GridLines="Horizontal">

<ItemTemplate>

id:

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

<br />

name:

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

<br />

score:

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

<br />

<br />

</ItemTemplate>

</asp:DataList>

</div>

</form>

</body>

</html>

File: WebForm1.aspx.cs

﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Practical6

{

public partial class WebForm1 : System.Web.UI.Page

{

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

{

}

protected void SqlDataSource1\_Selecting(object sender, SqlDataSourceSelectingEventArgs e)

{

}

protected void FormView1\_PageIndexChanging(object sender, FormViewPageEventArgs e)

{

}

}

}

File: WebForm2.aspx

﻿<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs" Inherits="Practical6.WebForm2" %>

<!DOCTYPE html>

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

<head runat="server">

<title>Practical 6-2</title>

</head>

<body>

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

<asp:GridView ID="GridView1" runat="server"></asp:GridView>

</form>

</body>

</html>

File: WebForm2.aspx.cs

﻿using System;

using System.Data;

using System.Data.SqlClient;

namespace Practical6

{

public partial class WebForm2 : System.Web.UI.Page

{

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

{

using (SqlConnection con = new SqlConnection("Data Source=DESKTOP-OUL5TLP\\SQLEXPRESS;Initial Catalog=coll;Integrated Security=True"))

{

con.Open();

SqlCommand cmd = new SqlCommand("ajayk57\_sql", con);

cmd.CommandType = CommandType.StoredProcedure;

SqlDataReader rdr = cmd.ExecuteReader();

GridView1.DataSource = rdr;

GridView1.DataBind();

}

}

}

}

File: WebForm3.aspx

﻿<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm3.aspx.cs" Inherits="Practical6.WebForm3" %>

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

Please Enter your information:<br />

Roll No.:

<asp:TextBox ID="TextBox1" runat="server" OnTextChanged="TextBox1\_TextChanged"></asp:TextBox>

<br />

Name:

<asp:TextBox ID="TextBox2" runat="server" OnTextChanged="TextBox1\_TextChanged"></asp:TextBox>

<br />

Score:

<asp:TextBox ID="TextBox3" runat="server" OnTextChanged="TextBox1\_TextChanged"></asp:TextBox>

<br />

<asp:Button ID="RegBtn" runat="server" Text="Register" OnClick="RegBtn\_Click" />

<br />

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

</div>

</form>

</body>

</html>

File: WebForm3.aspx.cs

﻿using System;

using System.Data;

using System.Data.SqlClient;

namespace Practical6

{

public partial class WebForm3 : System.Web.UI.Page

{

static StudentService StudentService;

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

{

StudentService = new StudentService();

}

protected void TextBox1\_TextChanged(object sender, EventArgs e)

{

}

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

{

//SqlConnection conn = new SqlConnection("Data Source=DESKTOP-OUL5TLP\\SQLEXPRESS;Initial Catalog=coll;Integrated Security=True");

//SqlCommand cmd = new SqlCommand("ajayk57\_sql2", conn);

//cmd.CommandType = CommandType.StoredProcedure;

//SqlParameter sqlpId = new SqlParameter("@id", SqlDbType.Int);

//SqlParameter sqlpName = new SqlParameter("@Name", SqlDbType.VarChar);

//SqlParameter sqlpScore = new SqlParameter("@score", SqlDbType.Int);

//sqlpId.Value = TextBox1.Text;

//sqlpName.Value = TextBox2.Text;

//sqlpScore.Value = TextBox3.Text;

//cmd.Parameters.Add(sqlpId);

//cmd.Parameters.Add(sqlpName);

//cmd.Parameters.Add(sqlpScore);

//conn.Open();

//cmd.ExecuteNonQuery();

//Label1.Text = "Succeeded";

//Console.WriteLine("huh?");

//Label1.Text = "Failed";

if (StudentService.AddStudentEntry(TextBox1.Text, TextBox2.Text, TextBox3.Text))

{

Label1.Text = "Succeeded";

}

else

{

Label1.Text = "Failed";

}

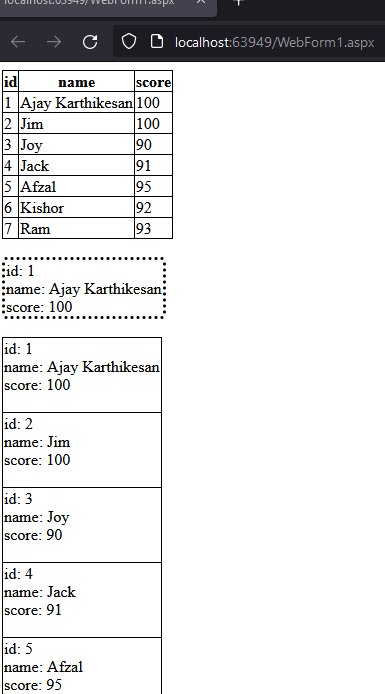
}

}

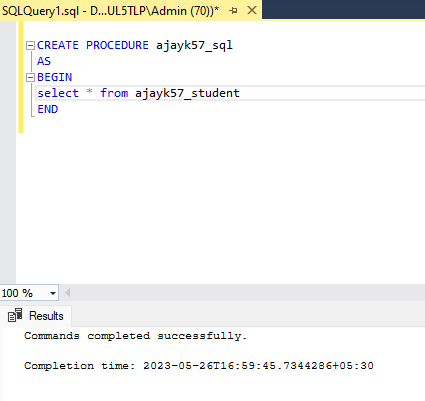
}

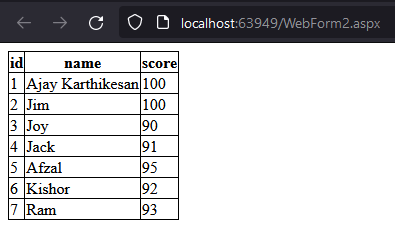
**Output:**

Q1)

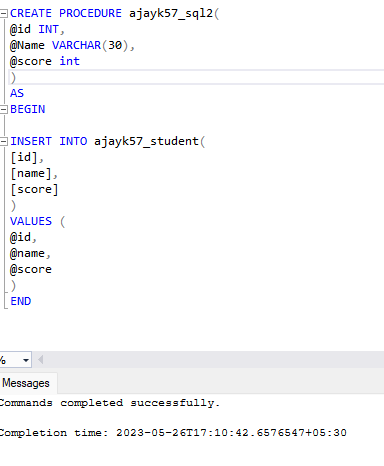


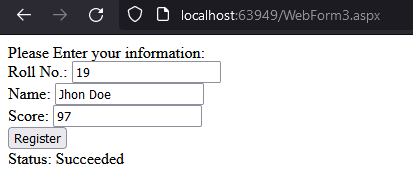
Q2)

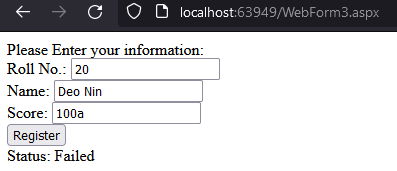




Q3)







**Conclusion:**

Successfully designed webpage to demonstrate use

of data bound controls, working of simple stored procedure and

parameterized stored procedure.