Aim: Create a simple Web application to Display your Details entered in registration form using advanced controls.

CODE:

registration.aspx

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

<!DOCTYPE html>

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

<head runat="server">

<title></title>

<style type="text/css">

.auto-style2 {

width: 41%;

height: 267px;

}

</style>

</head>

<body>

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

<div>

<br />

<table class="auto-style2">

<tr>

<td>

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

</td>

<td>:</td>

<td>

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

</td>

</tr>

<tr>

<td>

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

</td>

<td>:</td>

<td>

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

</td>

</tr>

<tr>

<td>

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

</td>

<td>:</td>

<td>

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

</td>

</tr>

<tr>

<td>

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

</td>

<td>:</td>

<td>

<asp:DropDownList ID="DropDownList1" runat="server">

<asp:ListItem>Mumbai</asp:ListItem>

<asp:ListItem>Pune</asp:ListItem>

<asp:ListItem>Ratnagiri</asp:ListItem>

<asp:ListItem>Raigad</asp:ListItem>

</asp:DropDownList>

</td>

</tr>

<tr>

<td>

<asp:Label ID="Label5" runat="server" Text="Gender"></asp:Label>

</td>

<td>:</td>

<td>

<asp:RadioButton ID="RadioButton1" runat="server" GroupName="gen" Text="Male" />

<asp:RadioButton ID="RadioButton2" runat="server" GroupName="gen" Text="Female"/>

</td>

</tr>

<tr>

<td>

<asp:Label ID="Label6" runat="server" Text="Gmail"></asp:Label>

</td>

<td>:</td>

<td>

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

</td>

</tr>

<tr>

<td colspan="3">

<asp:Button ID="Button1" runat="server" Text="Button" OnClick="Button1\_Click" />

</td>

</tr>

<tr>

<td colspan="3">

<asp:Label ID="Label7" runat="server" Text=""></asp:Label>

</td>

</tr>

</table>

</div>

</form>

</body>

</html>

**registration.aspx.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace RegistrationForm {

public partial class registration:System.Web.UI.Page {

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

}

protected void Button1\_Click(object sender,EventArgs e) {

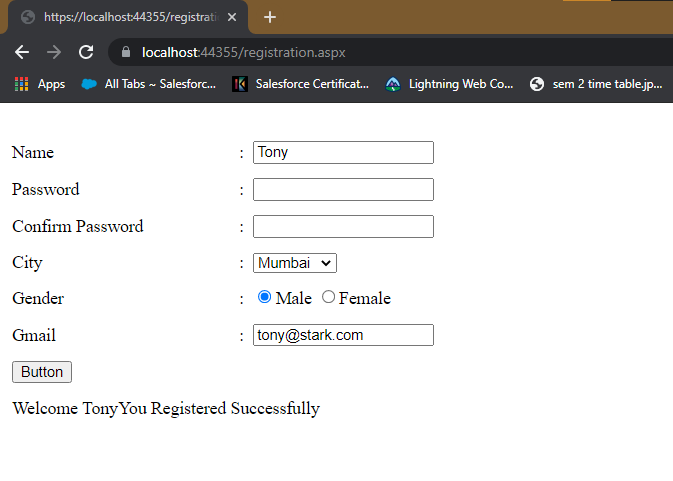
Label7.Text = "Welcome " + TextBox1.Text + "You Registered Successfully";

}

}

}

OUTPUT:



AIM: Develop a windows application to create StudentInfo (Name, RollNo, Sem) table & add, delete, modify, search records(Connected Architecture)

CODE:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace StudentInfo {

public partial class Form1:Form {

SqlConnection conn = new SqlConnection();

SqlDataReader dr;

SqlCommand cmd;

public Form1() {

InitializeComponent();

conn.ConnectionString = "Data Source=(localdb)\\MyInstance;Initial Catalog=master;Integrated Security=True";

}

private void button1\_Click(object sender,EventArgs e) {

label4.Text = "";

label5.Text = "";

label6.Text = "";

//int roll = Convert.ToInt16(textBox2.Text);

conn.Open();

cmd = new SqlCommand("Select \* from Studentinfo where roll =" + textBox2.Text);

cmd.Connection = conn;

dr = cmd.ExecuteReader();

while (dr.Read()) {

label4.Text += dr[0].ToString() + "\n";

label5.Text += dr[1].ToString() + "\n";

label6.Text += dr[2].ToString() + "\n";

}

conn.Close();

}

private void button2\_Click(object sender,EventArgs e) {

conn.Open();

cmd = new SqlCommand("insert into StudentInfo values('" + textBox1.Text + "'," + textBox2.Text + "," + textBox3.Text + ")");

cmd.Connection = conn;

cmd.ExecuteNonQuery();

conn.Close();

MessageBox.Show("Data Inserted Successfully!!!");

}

private void button3\_Click(object sender,EventArgs e) {

conn.Open();

cmd = new SqlCommand("update StudentInfo set Name='" + textBox1.Text + "',Sem=" + textBox3.Text + " where roll= " + textBox2.Text);

cmd.Connection = conn;

cmd.ExecuteNonQuery();

conn.Close();

MessageBox.Show("Data Updated Successfully!!!");

}

private void button4\_Click(object sender,EventArgs e) {

conn.Open();

cmd = new SqlCommand("delete from StudentInfo where roll = " + textBox2.Text);

cmd.Connection = conn;

cmd.ExecuteNonQuery();

conn.Close();

MessageBox.Show("Data Deleted Successfully!!!");

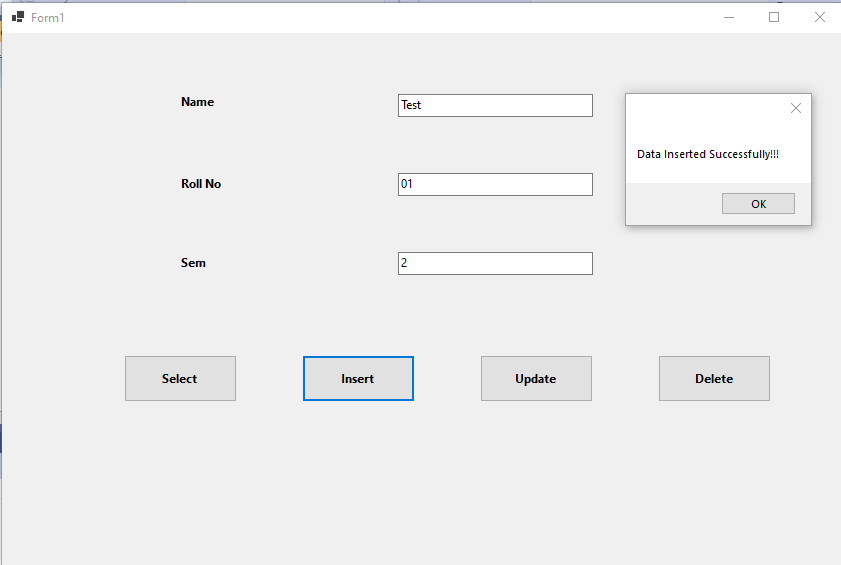
}

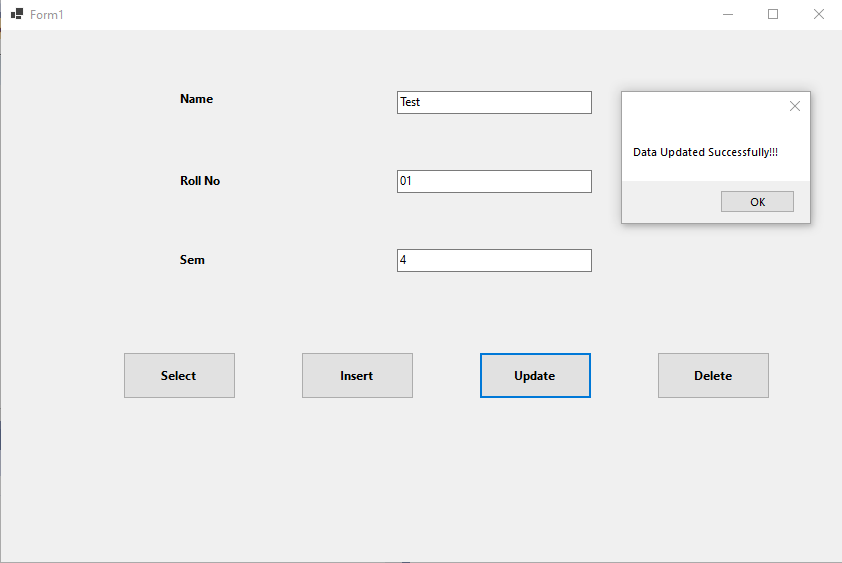
private void Form1\_Load(object sender,EventArgs e) {

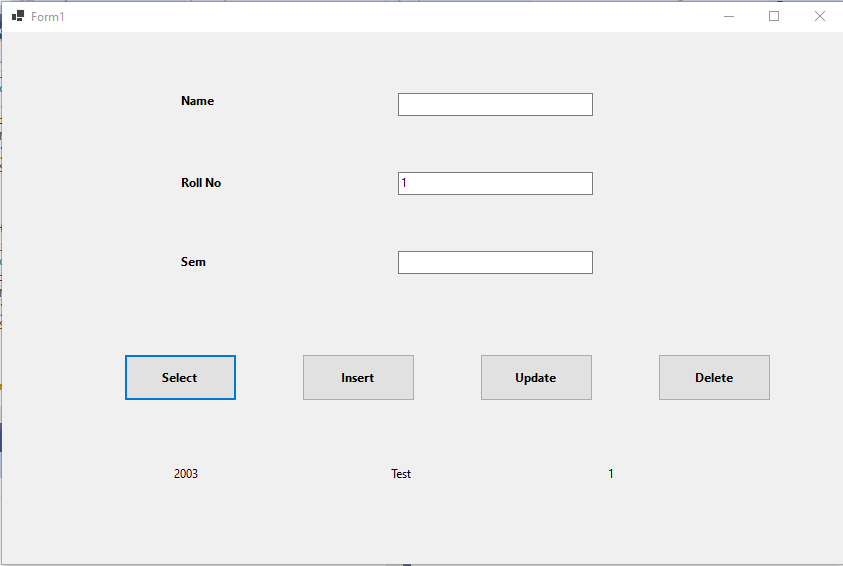
}

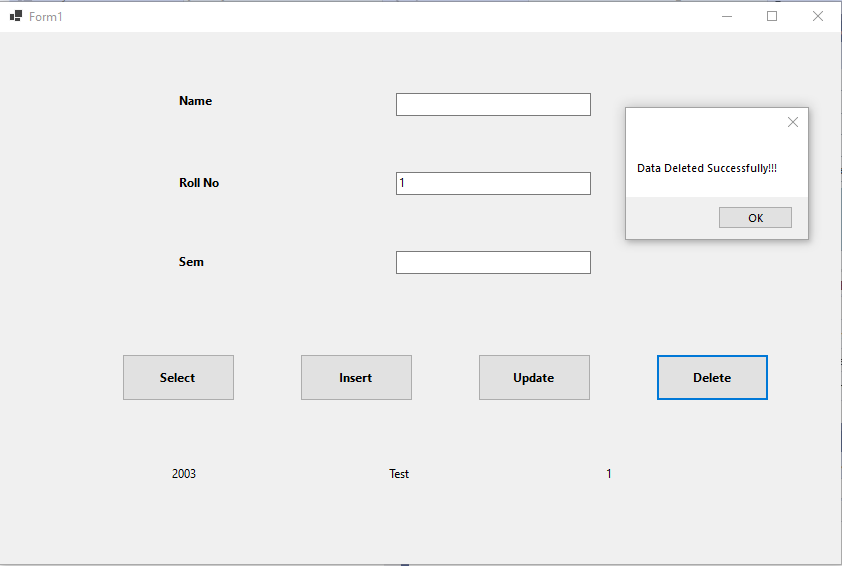
}

}









AIM: Design a program for multiple choice quiz application. Store at least 3 questions with their 4 optional answers, the correct answer & marks assigned in sql database table. Calculate total score & display result in message box on click of submit button. (Disconnected Architecture)

CODE:

Form1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Quizz\_App

{

public partial class Form1 : Form

{

SqlConnection con = new SqlConnection();

public Form1()

{

InitializeComponent();

con.ConnectionString = "Data Source=(localdb)\\MyInstance;Initial Catalog=master;Integrated Security=True";

con.Open();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

string q = "select ans from quizz";

SqlCommand cm = new SqlCommand(q, con);

SqlDataAdapter da = new SqlDataAdapter(cm);

DataTable dt = new DataTable();

da.Fill(dt);

int count = 0;

foreach (DataRow row in dt.Rows)

{

if (radioButton2.Checked)

{

if (radioButton2.Text == row["ans"].ToString())

{ count++; }

}

if (radioButton6.Checked)

{

if (radioButton6.Text == row["ans"].ToString())

{ count++; }

}

if (radioButton9.Checked)

{

if (radioButton9.Text == row["ans"].ToString())

{ count++; }

}

if (radioButton16.Checked)

{

if (radioButton16.Text == row["ans"].ToString())

{ count++; }

}

}

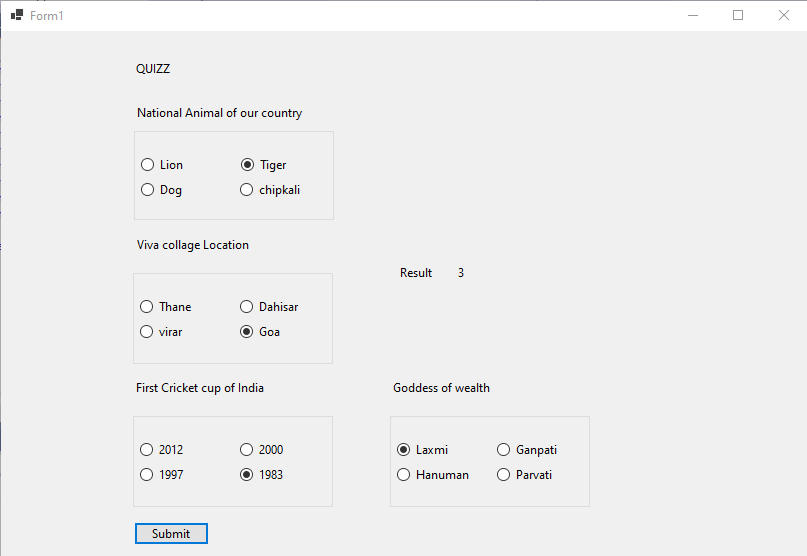
label7.Text = count.ToString();

}

}

}

Output:



AIM: Write a program on LinqToSQLClass. Show select, selectWith, insert, update, delete command on Course\Product Table.

Code :

ProductData.aspx

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

<!DOCTYPE html>

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

<head runat="server">

<title>Product</title>

<style type="text/css">

.auto-style2 {

width: 229px;

}

.auto-style3 {

width: 118px;

}

.auto-style4 {

width: 420px;

}

</style>

</head>

<body>

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

<div>

<br />

<br />

<table>

<tr>

<td class="auto-style2">Enter Product\_Name</td>

<td class="auto-style3">:</td>

<td class="auto-style4"><asp:TextBox ID="TextBox1" runat="server"></asp:TextBox></td>

</tr>

<tr>

<td class="auto-style2">Enter Product\_Price</td>

<td class="auto-style3">:</td>

<td class="auto-style4"><asp:TextBox ID="TextBox2" runat="server"></asp:TextBox></td>

</tr>

<tr>

<th class="auto-style2"><asp:Button ID="Button1" runat="server" Text="Add" OnClick="Button1\_Click1" /></th>

<th class="auto-style3"><asp:Button ID="Button2" runat="server" Text="Select" OnClick="Button2\_Click" /></th>

<th class="auto-style4">

<asp:Button ID="Button4" runat="server" OnClick="Button4\_Click" Text="Update" />

&emsp;

<asp:Button ID="Button3" runat="server" Text="Delete" OnClick="Button3\_Click" /></th>

</tr>

<tr>

<th class="auto-style2">

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

</th>

<th class="auto-style3">&nbsp;</th>

<th class="auto-style4">&nbsp;</th>

</tr>

</table>

<br />

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False" DataKeyNames="ID" DataSourceID="SqlDataSource1">

<Columns>

<asp:BoundField DataField="ID" HeaderText="ID" ReadOnly="True" SortExpression="ID" InsertVisible="False" />

<asp:BoundField DataField="P\_Name" HeaderText="P\_Name" SortExpression="P\_Name" />

<asp:BoundField DataField="P\_Price" HeaderText="P\_Price" SortExpression="P\_Price" />

</Columns>

</asp:GridView>

<br />

<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:masterConnectionString %>" SelectCommand="SELECT \* FROM [Product]"></asp:SqlDataSource>

<br />

<br />

</div>

</form>

</body>

</html>

ProductData.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace LinqToSql {

public partial class ProductData:System.Web.UI.Page {

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

}

protected void Button1\_Click1(object sender,EventArgs e) {

var d = new DataClassesDataContext();

var c = new Product();

c.P\_Name = TextBox1.Text;

c.P\_Price = Convert.ToInt16(TextBox2.Text);

d.Products.InsertOnSubmit(c);

d.SubmitChanges();

Label1.Text = "Successfully Added Data";

}

protected void Button2\_Click(object sender,EventArgs e) {

GridView1.DataBind();

}

protected void Button3\_Click(object sender,EventArgs e) {

var d = new DataClassesDataContext();

Product c = d.Products.FirstOrDefault(e1 => e1.P\_Name.Equals(TextBox1.Text));

c.P\_Name = TextBox1.Text;

d.Products.DeleteOnSubmit(c);

d.SubmitChanges();

Label1.Text = "Successfully Deleted Data";

GridView1.DataBind();

}

protected void Button4\_Click(object sender,EventArgs e) {

var d = new DataClassesDataContext();

Product c = d.Products.FirstOrDefault(e1 => e1.P\_Name.Equals(TextBox1.Text));

c.P\_Name = TextBox1.Text;

c.P\_Price = Convert.ToInt16(TextBox2.Text);

d.SubmitChanges();

Label1.Text = "Successfully Updated Data";

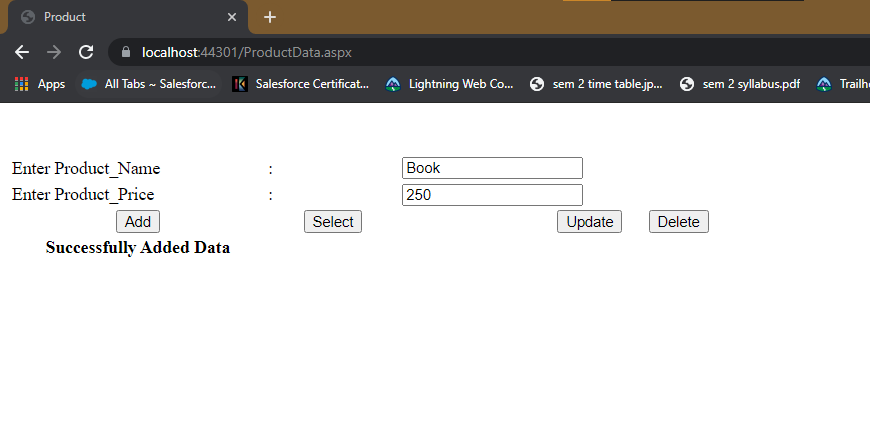
GridView1.DataBind();

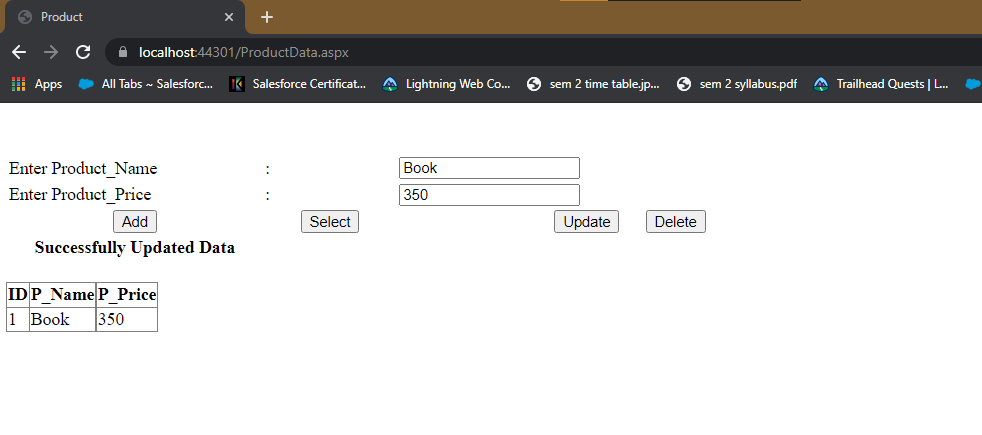
}

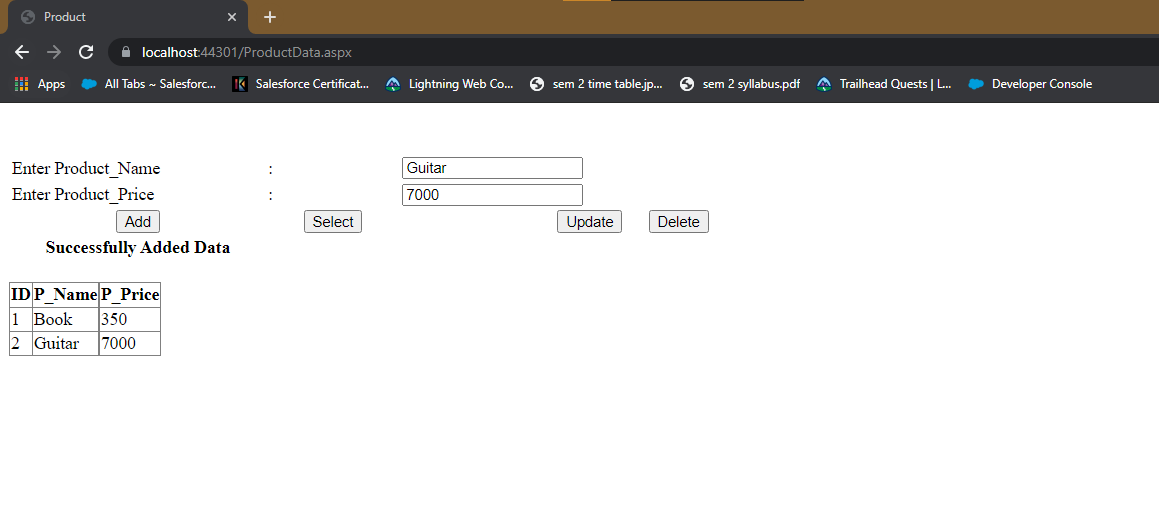
}

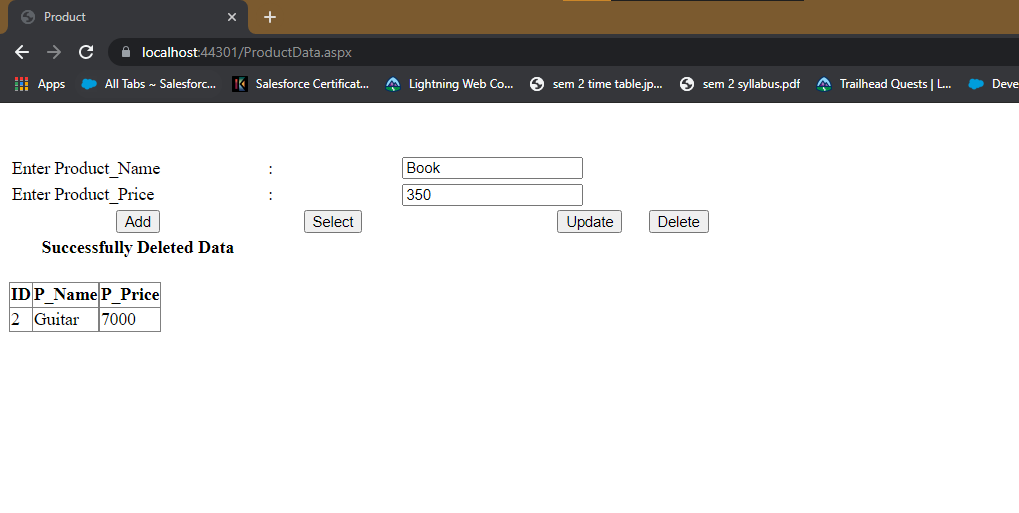
}

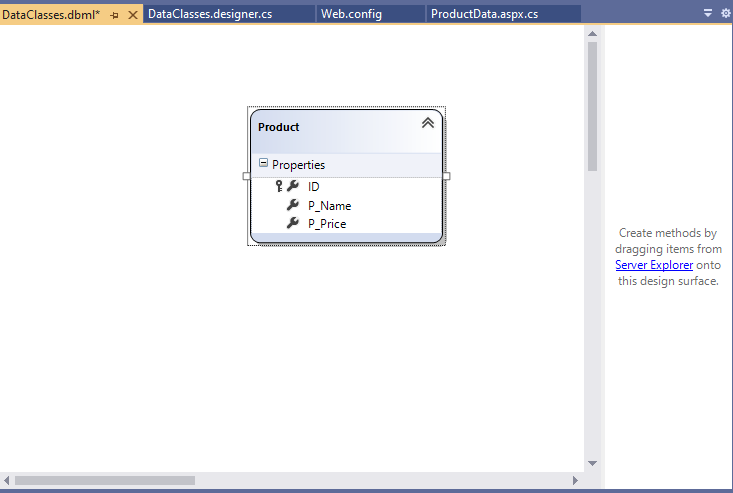
OUTPUT











AIM: Create ASP.NET program to demonstrate binding of different controls from database and display those records on grid view.

CODE :

TableList.aspx

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

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<br />

<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:masterConnectionString %>" SelectCommand="Select name from sys.tables"></asp:SqlDataSource>

<br />

<br />

List of Table :

<asp:DropDownList ID="DropDownList1" runat="server" DataSourceID="SqlDataSource1" DataTextField="name" DataValueField="name">

</asp:DropDownList>

<br />

<br />

<br />

<br />

<asp:Button ID="Button1" runat="server" OnClick="Button1\_Click" Text="Show Data" />

<br />

<br />

&nbsp;<br />

<br />

<asp:SqlDataSource ID="SqlDataSource2" runat="server"></asp:SqlDataSource>

<br />

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

</asp:GridView>

</div>

</form>

</body>

</html>

TableList.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.SqlClient;

using System.Data;

namespace DataBinding {

public partial class TableList:System.Web.UI.Page {

static string conn = System.Configuration.ConfigurationManager.ConnectionStrings["masterConnectionString"].ConnectionString;

SqlConnection cn = new SqlConnection(conn);

SqlCommand cmd;

SqlDataReader dr;

SqlDataAdapter adapt;

DataTable dt;

DataSet ds;

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

}

protected void Button1\_Click(object sender,EventArgs e) {

String q = "select \* from " + DropDownList1.Text;

cn.Open();

adapt = new SqlDataAdapter(q,cn);

dt = new DataTable();

adapt.Fill(dt);

GridView1.DataSource = dt;

cn.Close();

GridView1.DataBind();

}

}

}

Output:

