**Part B: Ex No: 1- To implement a Text Editor with Cut, Copy, Paste, Close and Save**

**Aim:**

Develop a menu based VB.Net application to implement a text editor with cut, copy, paste, save and close operations.

**Pre-Learning Skills**

* Understanding the menu based application
* Knowledge in basic tools in text editor

**Procedure:**

1. Open a New project.
2. Choose the **windows** **application** from the Templates.
3. Design the Form as follows
4. Change the Form Caption to the Title of the program
5. Drag and Drop required Controls from the tool box and change the name and text properties.
6. Add the menu Properties.
7. Write the following code in the click () event of the Menu option.
8. Save and Execute.

**Source Code:**

1: using System;

2: using System.Windows.Forms;

3:

4: namespace Ex\_8

5: {

6: public partial class Form1 : Form

7: {

8: public Form1()

9: {

10: InitializeComponent();

11: }

12: private void copyToolStripMenuItem\_Click()

13: {

14: richTextBox1.Copy();

15: }

16: private void pasteToolStripMenuItem\_Click()

17: {

18: richTextBox1.Paste();

19: }

20: private void cutToolStripMenuItem\_Click()

21: {

22: richTextBox1.Cut();

23: }

24: private void clearToolStripMenuItem\_Click()

25: {

26: richTextBox1.Clear();

27: }

28: private void closeToolStripMenuItem\_Click()

29: {

30: this.Close();

31: }

32: private void newToolStripMenuItem\_Click()

33: {

34: richTextBox1.Clear();

35: }

36: private void openToolStripMenuItem\_Click()

37: {

38: openFileDialog1.ShowDialog();

39: richTextBox1.LoadFile(openFileDialog1.FileName,

40: RichTextBoxStreamType.PlainText);

41: }

42: private void saveToolStripMenuItem\_Click()

43: {

44: saveFileDialog1.ShowDialog();

45: richTextBox1.SaveFile(saveFileDialog1.FileName,

46: RichTextBoxStreamType.PlainText);

47: }

48: private void fontToolStripMenuItem\_Click()

49: {

50: fontDialog1.ShowDialog();

51: richTextBox1.Font = fontDialog1.Font;

52: }

53: private void colorToolStripMenuItem\_Click()

54: {

55: colorDialog1.ShowDialog();

56: richTextBox1.ForeColor = colorDialog1.Color;

57: }

58: }

59: }

**Output:** (Print your screen for output)

**Result :**

The menu based application was designed with menustrip and dialog control and the program was successfully executed and then the output was verified.

**Viva Questions**

1. **How to assign FONT to label during runtime?**

Label.font.name=”Times New Roman”;

Label.font.size=20

1. **What are common dialog controls?**

The common dialog boxes control, which is a custom control, allows your project to use the dialog boxes that are provided as a part of the Windows environment.

1. **Name some showprinter method in Printer object?**

cdlPrinter.ShowPrinter  
Printer.Copies = cdlPrinter.Copies  
Printer.Orientation = cdlPrinter.Orientation  
Printer.EndDoc

1. Which dialog control allows the user to zoom in on a document?

a.) PrintDialog

b.) PrintPreview

c.) PageSetupDialog

d.) Both a and b.

**Skill Practice;**

* List the Dialog Boxes used in Windows Forms of VB.NET.

|  |  |  |
| --- | --- | --- |
| **Dialogbox Control Name** | **Properties** | **Method Used** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Part B - Ex No: 2 - To perform a quiz of 5 questions using**

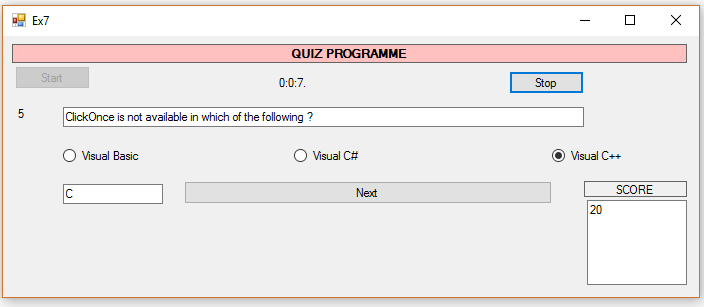
**timer control**

**Aim:**

To develop a VB.Net application to perform timer based quiz of 5 questions.

**Procedure:**

1. Open a New project.
2. Choose the **windows** **application** from the Templates.
3. Design the Form as follows

****

1. Change the Form Caption to the Title of the program
2. Drag and Drop required Controls from the tool box and change the name and text properties.
3. Place a button to start the quiz program.
4. To view next question place a button NEXT
5. Place a Timer Control and write coding on the Tick\_Event() for timing.
6. Write the following code in the click () event of all the buttons.
7. Save and Execute (press F5).

**Source Code:**

1: using System;

2: using System.Windows.Forms;

3:

4: namespace Ex\_9

5: {

6: public partial class Form1 : Form

7: {

8: public Form1()

9: {

10: InitializeComponent();

11: }

12: int score = 0;

13: int i = -1;

14: int a = 0;

15: DateTime startTime;

16: string[] questions = new string[]

17: {

18: "The type float can be safely converted

to\_\_\_?",

19: "Default Property for a textbox

control\_\_\_\_\_\_?",

20: "What is the operator used to create object?",

21: "\_\_\_\_\_\_\_\_loop repeats group of statement for

each element of an array?",

22: "A \_\_\_\_\_\_\_\_\_variable is one that is declared

inside a method?"

23:

24: };

25: string[] answers = new string[]

26: {

27: "double", "long", "decimal", "ufloat",

28: "Multiline", "Password Char", "Enable", "Text",

29: "++", "+", "New", "object",

30: "while loop", "do while loop", "foreach loop",

31: "All the Above",

32: "global","local","external","static"

33: };

34: string[] quizAnswers = new string[] { "double",

"Text", "New", "foreach loop", "local" };

35: string getSelectedAnswer()

36: {

37: if (rdb\_A.Checked)

38: return rdb\_A.Text.ToString();

39: if (rdb\_B.Checked)

40: return rdb\_B.Text.ToString();

41: if (rdb\_C.Checked)

42: return rdb\_C.Text.ToString();

43: if (rdb\_D.Checked)

44: return rdb\_D.Text.ToString();

45: return "";

}

46: public void check\_answer()

47: {

48: if (getSelectedAnswer().Equals(quizAnswers[i]))

49: {

50: MessageBox.Show("Correct");

51: score++;

52: txt\_score.Text = Convert.ToString(score);

53: btn\_next\_question.Enabled = false;

54: btn\_next\_question.Visible = false;

55: btn\_start.Visible = true;

56: btn\_start.Enabled = true;

57: // btn\_start.Text = "Next";

58: }

59: else

60: {

61: MessageBox.Show("Incorrect");

62: // score--;

63: txt\_score.Text = Convert.ToString(score);

64: btn\_next\_question.Enabled = false;

65: btn\_next\_question.Visible = false;

66: btn\_start.Visible = true;

67: btn\_start.Enabled = true;

68: // btn\_start.Text = "Next";

69: }

70: }

71: public void load\_question()

72: {

73: startTime = DateTime.Now;

74: i++;

75: if (i < questions.Length)

76: {

77: //txtScore.Text = score;

78: txt\_question\_no.Text = (i + 1).ToString();

79: txt\_question.Text = questions[i];

80:

81: rdb\_A.Text = answers[a];

82: a++;

83: rdb\_B.Text = answers[a];

84: a++;

85: rdb\_C.Text = answers[a];

86: a++;

87: rdb\_D.Text = answers[a];

88: a++;

89:

90: rdb\_A.Checked = false;

91: rdb\_B.Checked = false;

92: rdb\_C.Checked = false;

93: rdb\_D.Checked = false;

94:

95: btn\_start.Enabled = false;

96: // btn\_next\_question.Visible = true;

97: // btn\_next\_question.Enabled = true;

98: timer1.Start();

99: }

100: }

101: private void btn\_next\_question\_Click()

102: {

103: check\_answer();

104: }

105: private void timer1\_Tick(object sender, EventArgs e)

106: {

107: TimeSpan span = DateTime.Now.Subtract(startTime);

108: txt\_time.Text = span.Hours.ToString() + ":" +

109: span.Minutes.ToString() + ":" +

span.Seconds.ToString() + ".";

110: //& span.Milliseconds

111: //if (span.Minutes == 1) // sets for 1 minute

112: if (span.Seconds == 5) // sets for 5 milliseconds

113: {

114: timer1.Stop();

115: check\_answer();

116: load\_question();

117: }

118: }

119: private void btn\_start\_Click()

120:

121: {

122: startTime = DateTime.Now;

123: load\_question();

124: }

125: }

126: }

**Output:** (Print your screen for output)

**Result:**

The windows application is created and designed for a Quiz event with 5 questions. The program code was executed with timer control and the scores were calculated and displayed.

**Viva Questions and Answers:**

1. **Difference between Panel and Group Box classes?**

Panel and Group box both can used as container for other controls like radio buttons and check box.

The difference in panel and group box is Panel

* In case of panel captions cannot be displayed
* Can have scroll bars. Group box
* Captions can be displayed.
* Cannot have a scroll bar

1. **What is the difference between “continue” and “break” statement in C#?**

Using break statement, you can 'jump out of a loop' whereas by using continue statement, you can 'jump over one iteration' and then resume your loop execution.

1. **What is Timer Control?**

Timer Control allows us to set Interval property in milliseconds like (1 second equal to 1000 milliseconds). For example, if we want to set an interval of one minute we set the value at the Interval property as 60000, which means 60×1000.

The Timer Control runs only when its Enabled property is set True, by default Enabled property is always False.

**Part B - Ex No: 3 - To insert, update, delete operation using**

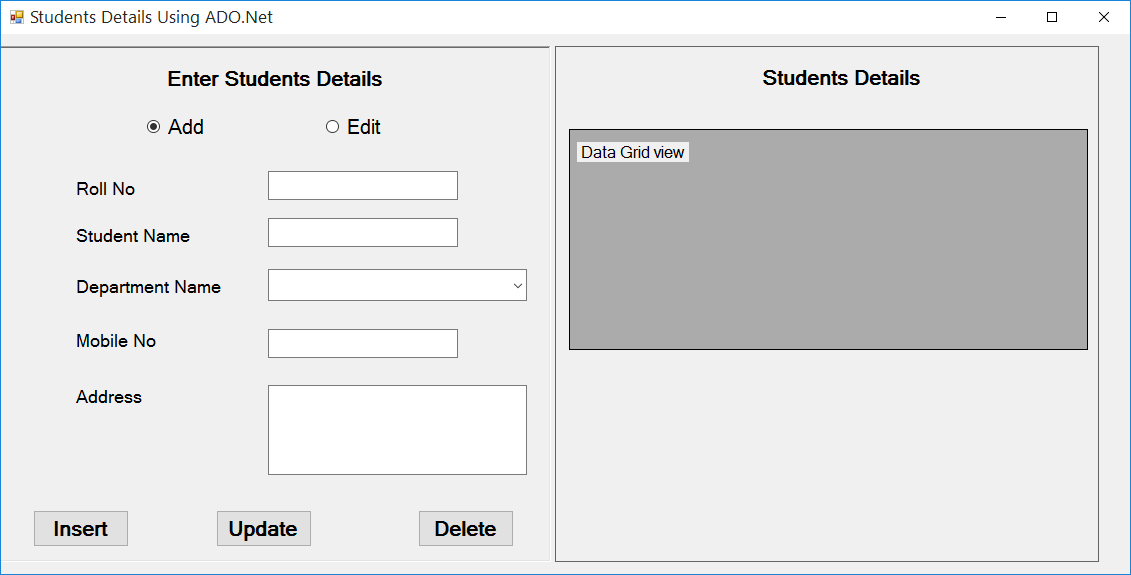
**ADO.Net**

**Aim:**

To develop a database application using ADO.NET to insert, modify, update and delete operations.

**Procedure:**

1. Open a New project.
2. Choose the **windows** **application** from the Templates.
3. Design the Form as follows



* Change the Form Caption to the Title of the program
* Drag and Drop required Controls from the tool box and change the name and text properties.

1. Write the following code in the click () event of all the buttons.

1. Open the ADO.Net Connection

2. Execute Query

3. Close the Connection

1. Save and Execute (press F5).

**Source Code:**

1: Using System;

2: using System.Data;

3: using System.Data.SqlClient;

4: using System.Windows.Forms;

5:

6: namespace Ex\_10

7: {

8: public partial class Form1 : Form

9: {

10: public Form1()

11: {

12: InitializeComponent();

13: }

14: //CONNECTION OBJECT

15: SqlConnection con = new SqlConnection();

16: // COMMAND OBJECT

17: SqlCommand cmd = new SqlCommand();

18: string sql;

19: private void rdb\_Edit\_CheckedChanged()

20: {

21: cmb\_Rno.Visible = true;

22: txt\_rno.Visible = false;

23: btn\_save.Text = "Update";

24: clear();

25: LoadData();

26: }

27: private void rdb\_add\_CheckedChanged()

28: {

29: cmb\_Rno.Visible = false;

30: txt\_rno.Visible = true;

31: btn\_save.Text = "Save";

32: clear();

33: }

34: private void rdb\_delete\_CheckedChanged()

35: {

36: cmb\_Rno.Visible = true;

37: txt\_rno.Visible = false;

38: btn\_save.Text = "Delete";

39: clear();

40: LoadData();

41: }

42: private void rdb\_select\_CheckedChanged()

43: {

44: cmb\_Rno.Visible = false;

45: txt\_rno.Visible = true;

46: rdb\_add.Checked = false;

47: btn\_save.Text = "Add";

48: BindData();

49: }

50: private void btn\_save\_Click()

51: {

52: int x;

53: connect();

54: switch (btn\_save.Text)

55: {

56: case "Save":

57: sql = "insert into

student\_master(student\_rno,student\_name,dept\_name,address,mobile\_no)values(@rno,@stuname,@dept,@address,@mobileno)";

58: cmd = new SqlCommand(sql, con);

59: cmd.Parameters.Add("@rno",

SqlDbType.Int).Value = txt\_rno.Text;

cmd.Parameters.Add("@stuname",

SqlDbType.VarChar).Value =

txt\_stuname.Text;

60: cmd.Parameters.Add("@dept",

61: SqlDbType.VarChar).Value = cmb\_deptname.SelectedItem.ToString();

62: cmd.Parameters.Add("@address",

SqlDbType.VarChar).Value =

txt\_address.Text;

63: cmd.Parameters.Add("@mobileno",

SqlDbType.VarChar).Value = txt\_mobile.Text;

64: x = cmd.ExecuteNonQuery();

65: if (x == 1)

66: MessageBox.Show("Students Data Inserted");

67: else

68: MessageBox.Show("Students Data Not

Inserted");

69: con.Close();

70: clear();

71: break;

72: case "Update":

73: sql = "update student\_master set

student\_name=@stuname,dept\_name=@dept,address=@address,mobile\_no=@mobileno where student\_rno=@rno";

74: cmd = new SqlCommand(sql, con);

75: cmd.Parameters.Add("@rno",

SqlDbType.VarChar).Value =

cmb\_Rno.SelectedItem.ToString();

76: cmd.Parameters.Add("@stuname",

SqlDbType.VarChar).Value =

txt\_stuname.Text;

77: cmd.Parameters.Add("@dept",

SqlDbType.VarChar).Value =

cmb\_deptname.SelectedItem.ToString();

78: cmd.Parameters.Add("@address",

SqlDbType.VarChar).Value = txt\_address.Text;

79: cmd.Parameters.Add("@mobileno",

SqlDbType.VarChar).Value = txt\_mobile.Text;

80: x = cmd.ExecuteNonQuery();

81: if (x == 1)

82: MessageBox.Show("Students Data

Update");

83: else

84: MessageBox.Show("Students Data Not

Updated");

85: con.Close();

86: clear();

87: break;

88: case "Delete":

89: sql = "delete from student\_master where

student\_rno=@rno";

90: cmd = new SqlCommand(sql, con);

91: cmd.Parameters.Add("@rno",

SqlDbType.VarChar).Value =

cmb\_Rno.SelectedItem.ToString();

92:

93: x = cmd.ExecuteNonQuery();

94: if (x == 1)

95: MessageBox.Show("Selected Student Data

Deleted");

96: else

MessageBox.Show("Selected Student Data Not

Deleted");

97: con.Close();

98: clear();

99: LoadData();

100: break;

101:

102: }

103: BindData();

104: }

105: public void connect()

106: {

107: openFileDialog1.ShowDialog();

108: con.ConnectionString = @"Data

Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=" +

openFileDialog1.FileName + @";Integrated

Security=True;Integrated Security=True";

109: con.Open();

110: }

111: public void clear()

112: {

113: txt\_rno.Clear();

114: txt\_stuname.Clear();

115: txt\_address.Clear();

116: txt\_mobile.Clear();

117: cmb\_Rno.Text = "";

118: cmb\_deptname.Text = "";

119: }

120: public void BindData()

121: {

122: // Call connect function

123: connect();

124: cmd.Connection = con;

125: cmd.CommandText = "select \* from student\_master ";

126: DataSet ds = new DataSet();

127: SqlDataAdapter ada = new SqlDataAdapter();

128: ada.SelectCommand = cmd;

129: ada.Fill(ds, "student\_master");

130: dataGridView1.DataSource = ds;

131: dataGridView1.DataMember = "student\_master";

132: con.Close();

133: }

134: public void LoadData()

135: {

136: connect();

137: cmd.Connection = con;

138: cmd.CommandText = "select \* from student\_master ";

139: cmb\_Rno.Items.Clear();

140: //EXECUTION OF ADO

141: SqlDataReader dr = null;

142: dr = cmd.ExecuteReader();

143: while (dr.Read())

144: {

145: cmb\_Rno.Items.Add((dr.GetInt32(0)));

146: }

147: con.Close();

148: }

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

150: {

151: BindData();

152: }

153: private void cmb\_Rno\_SelectedIndexChanged()

154: {

155: connect();

156: cmd.Connection = con;

157: cmd.CommandText = "select \* from student\_master

where student\_rno = '" +

cmb\_Rno.SelectedItem.ToString() + "'";

158: SqlDataReader dr = null;

159: dr = cmd.ExecuteReader();

160: while (dr.Read())

161: {

162: txt\_stuname.Text = dr.GetString(1);

163: cmb\_deptname.Text = dr.GetString(2);

164: txt\_address.Text = dr.GetString(3);

165: txt\_mobile.Text = dr.GetString(4);

166: }

167: con.Close();

168: }

169: }

170: }

**Output:** (Print your screen for output)

**Result :**

The ADO.Net windows application was designed and executed the CRUD operations using back end SQL Server 2008 and the result were verified by displaying data in datagridview in front end.

**Viva Questions and Answers:**

1. **What are the namespaces used in ADO.Net to connect to a database?**

* The System.Data namespace.
* The System.Data.OleDb namespace – A data provider used to access database such as Access, Oracle, or SQL.
* The System.Data.SQLClient namespace – Used to access SQL as the data provider.

**Skill Practice;**

* Observe and fill out the format

|  |  |  |
| --- | --- | --- |
| **Control Name Used** | **Control Property Name** | **Control**  **Property Value** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **Methods Used** | | |
|  | | |
|  | | |
|  | | |
|  | | |

**Part B - Ex No: 4 - Data grid to Add, Edit and Modify Records**

**Aim:**

To develop a VB.Net application using Datagrid to add, edit and modify records.

**Procedure:**

1. Add Datagrid from toolbox to your windows form
2. Select Data Source property from the Properties window
3. Then click Add Project data Source
4. Select Database in the Data source configuration wizard and click next
5. Click New Connection
6. Browse for the Data source name in the Add connection Wizard
7. Press Test Connection for success connection
8. Click Next and Data set object window will be opened
9. Select the dataset Table to display and click Finish.
10. Save the windows application and Execute.

**Source Code:**

1: using System;

2: using System.Data;

3: using System.Data.SqlClient;

4: using System.Windows.Forms;

5:

6: namespace Ex\_11

7: {

8: public partial class Form1 : Form

9: {

10: public Form1()

11: {

12: InitializeComponent();

13: }

14: //CONNECTION OBJECT

15: SqlConnection con = new SqlConnection();

16: // COMMAND OBJECT

17: SqlCommand cmd = new SqlCommand();

18: SqlDataAdapter ada;

19: DataSet ds = new DataSet();

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

21: {

22: BindData();

23: }

24: public void connect()

25: {

26: openFileDialog1.ShowDialog();

27: con.ConnectionString = @"Data

Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=" +

openFileDialog1.FileName + @";Integrated

Security=True;Integrated Security=True";

28: con.Open();

29: }

30: public void BindData()

31: {

32: // Call connect function

33: connect();

34: ada = new SqlDataAdapter("select \* from

student\_master", con);

35: SqlCommandBuilder cmdb = new

SqlCommandBuilder(ada);

36: ada.Fill(ds, "student\_master");

37: dataGridView1.DataSource = ds;

38: dataGridView1.DataMember = "student\_master";

39: con.Close();

40: }

41: private void btn\_save\_Click()

42: {

43: int x;

44: x = ada.Update(ds, "student\_master");

45: if (x == 1)

46: MessageBox.Show("RECORD UPDATED");

47: else

48: MessageBox.Show("RECORD NOT UPDATED");

49: }

50: }

51: }

**Output:** (Print your screen for output)

**Result:**   
The above ADO.Net Program was created with datagridview control to add and edit student’s records and the program was successfully executed and the results were verified.

**Skill Practice;**

* Observe and fill out the format

|  |  |  |
| --- | --- | --- |
| **Control Name Used** | **Control Property Name** | **Control**  **Property Value** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **Methods Used** | | |
|  | | |
|  | | |

**Viva Questions and Answers:**

1. What is DataGrid?

DataGrid is Web server control a powerful tool for displaying information from a data source. We can display editable data in a professional-looking grid by setting only a few properties. The grid has a sophisticated object model that provides you with great flexibility in how you display the data.

1. **Difference between Datagrid and data reader?**

A datagrid is a simple [server control](http://www.newinterviewquestions.com/interview/8647/) to which data from a data source can be binded. Where as a data reader is a forward only stream of data from the [data source](http://www.newinterviewquestions.com/interview/8647/) that holds reference to one record at any given time

1. **What is the purpose of using DataGrid control?**

* The DataGrid control is used to display the fields of the table. It is basically associated with the data source that is present in the column of a table.
* DataGrid provides the access to the rows that allows the control to be taken to the record in the data source.
* DataGrid control provides various features like editing, selection, deleting, paging and sorting.

**Part B – Ex. No: 5 - Web Application with validation controls to**

**validate web data from database**

**Aim:**

Design a Web application with Required Field Validator and RangeValidator

Controls to input data through a web form to a database and validate the data.

**Procedure:**

* 1. Open Visual Studio –> Create a new empty Web application.
  2. Create a new web page and design web form with three textbox control along with button control.
  3. Drag and drop RangeValidator control from Toolbox.
  4. Set ControlToValidate and Text property of RangeValidator control
  5. Set MaximumValue and MinimumValue Property.
  6. Set Type property for Data type of values for comparison.
  7. Drag and drop RequiredFieldValidator control from Toolbox.
  8. Set ControlToValidate and Text property of RequiredFieldValidator control
  9. Set the property of ControlToValidate = ID of the control (textbox) to validate.
  10. Add scriptmanager from toolbox
  11. Enable the validation mode in the web.config file. Add key and value in the appsetting like as below

<appSettings>  
 <add key="ValidationSettings:UnobtrusiveValidationMode" value="None" />  
   </appSettings>

Note: If you set an unobtrusiveValidation mode to none(default), the asp.net application will use the pre 4.5 behavior for client side validation. If you set the key value to web forms, the application will use the HTML5 attributes for client side validation.

**Source Code:**(Web Design Page)

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="login\_validation.aspx.cs" Inherits="Ex\_12.login\_validation" %>

<!DOCTYPE html>

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

<head runat="server">

<title></title>

<style type="text/css">

.auto-style1 {

width: 334px;

}

.auto-style2 {

width: 350px;

}

</style>

</head>

<body>

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

<div>

<fieldset style="width:600px;">

<legend>Required Field Validator in asp.net</legend>

<table align="center" style="border: 1px solid #dbcece">

<tr>

<td colspan="3" style="text-align: center; font-

weight: 700; border-bottom-style: solid;border-

bottom-width: thin; border-bottom-

color:#008080;">User Login Area</td>

</tr>

<tr>

<td >&nbsp;</td>

<td >&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td >UserName :</td>

<td >

<asp:TextBox ID="txtusername" runat="server"

Width="120px"></asp:TextBox>

</td>

<td>

<asp:RequiredFieldValidator

ID="RequiredFieldValidator3" runat="server"

ControlToValidate="txtusername"

ErrorMessage="Please, enter username"

ForeColor="Red"></asp:RequiredFieldValidator>

</td>

</tr>

<tr>

<td >Password :</td>

<td >

<asp:TextBox ID="txtpassword" runat="server"

TextMode="Password" Width="120px">

</asp:TextBox>

</td>

<td>

<asp:RequiredFieldValidator

ID="RequiredFieldValidator4"

runat="server" ControlToValidate="txtpassword"

ErrorMessage="Please, enter password"

ForeColor="Red"></asp:RequiredFieldValidator>

</td>

</tr>

</table>

</fieldset>

</div>

<div>

<fieldset style="width:600px;">

<legend>RangeValidator in asp.net</legend>

<table align="center" style="border: 1px solid #dbcece" class="auto-style2">

<tr>

<td>Age <span style="color:red;">\*</span></td>

<td class="auto-style1">

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

<br />

<asp:RangeValidator ID="rgvAge" runat="server" ErrorMessage="Please enter age between 18 to 30" ForeColor="#FF3300" MaximumValue="30" MinimumValue="18" SetFocusOnError="True" Type="Integer" ControlToValidate="txtAge">

</asp:RangeValidator>

</td>

<td> </td>

</tr>

<tr>

<td>&nbsp;</td>

<td class="auto-style1">

<asp:Button ID="btnlogin" runat="server"

OnClick="btnlogin\_Click" Text="Register" />

</td>

<td> </td>

</tr>

</table>

</fieldset>

</div>

<asp:ScriptManager ID="ScriptManager1" runat="server">

</asp:ScriptManager>

</form>

</body>

</html>

**Source Code:**

1. using System;

2. using System.Web;

3. using System.Web.UI;

4. using System.Web.UI.WebControls;

5. using System.Data;

6. using System.Data.SqlClient;

7. namespace Ex\_12\_A

8. {

9. public partial class User\_register : System.Web.UI.Page

10. {

11. protected void btnlogin\_Click(object sender, EventArgs e)

12. {

13. SqlConnection con = new SqlConnection("Data

Source=(LocalDB)[\\MSSQLLocalDB;AttachDbFilename=E:\\labcode\_cbt\\CBT](file:///\\MSSQLLocalDB;AttachDbFilename=E:\\labcode_cbt\\CBT)

\_Lab\_N\\Ex\_12\_A\\App\_Data\\ex\_12A\_db.mdf;Integrated Security=True");

14. SqlCommand cmd = new SqlCommand("INSERT INTO

user\_register(User\_Name,Password,Age) VALUES(@name,@pass,@age)");

15. int x;

16. cmd.Connection = con;

17. cmd.Parameters.AddWithValue("@name", txtusername.Text);

18. cmd.Parameters.AddWithValue("@pass", txtpassword.Text);

19. cmd.Parameters.AddWithValue("@age", txtAge.Text);

20. con.Open();

21. x = cmd.ExecuteNonQuery();

22. if (x == 1)

23. Response.Write("<script>alert('Registered Successful...');

</script>");

24. else

25. Response.Write("<script>alert('Not Registered...');</script>");

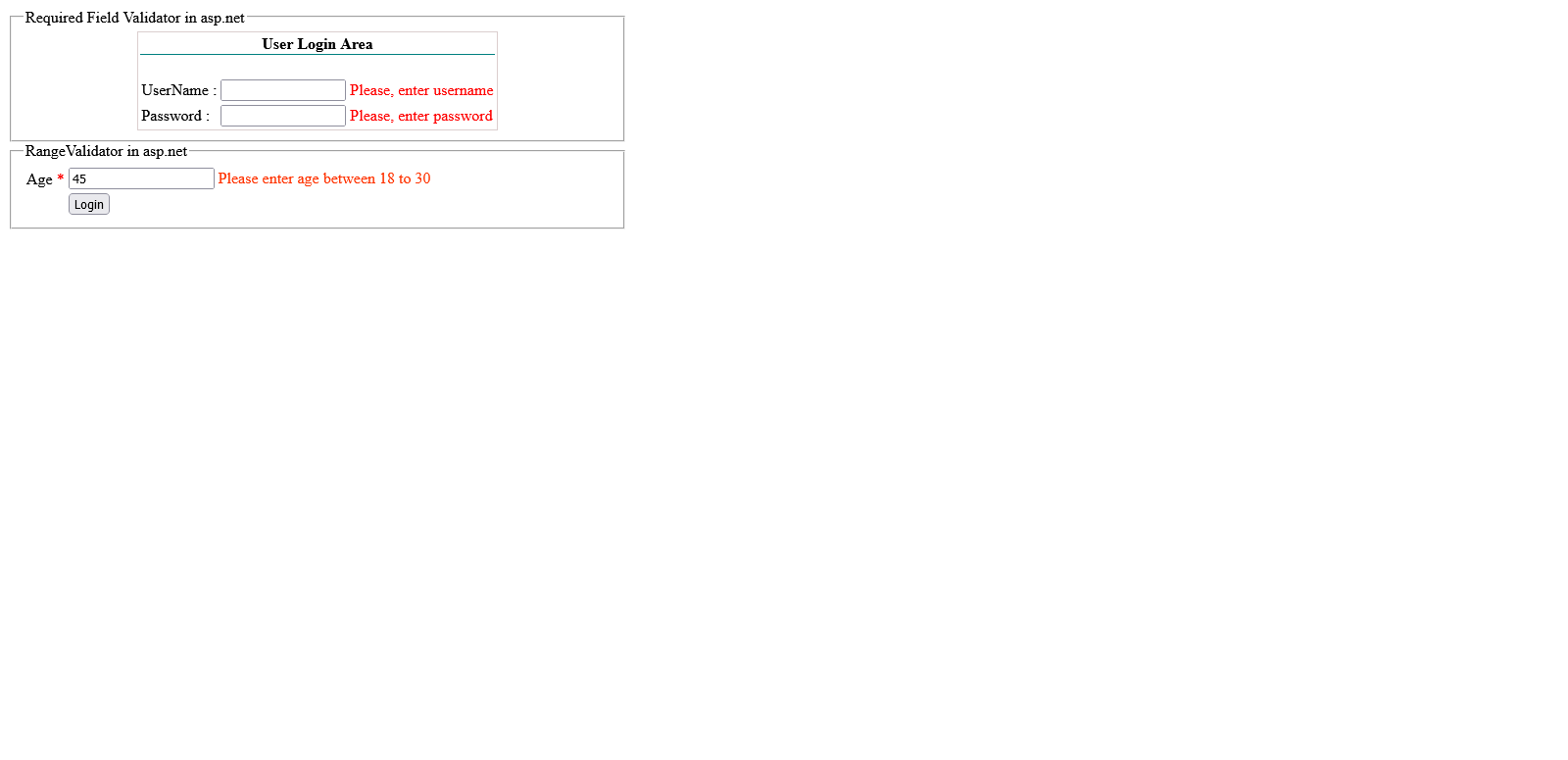
26. con.Close();

27. }

28. }

29. }

**Sample Output:**



**Result :**

The web application program was designed with validation control and validated for required values and range of values in client side by submitting the web page.

**Viva Questions and Answers:**

1. **What are the different validation controls available in ASP.Net?**  
   **RequiredFieldValidator:** Verifies whether a control contains data  
   **CompareValidator:** Verifies whether an entered item matches an entry in another control  
   **RangeValidator:** Verifies whether an entered item is between two values  
   **RegularExpressionValidator:** Verifies whether an entered item matches a specified format

**CustomValidator:** Verifies the validity of an entered item using a client-side script or a server-side code, or both

**ValidationSummary:** Displays validation errors in a central location or display a general validation error description.

1. **Where do the ASP.NET validation controls validate data, on the Client or on the Web Server?**

ASP.NET validation controls validate data on the client as well as web server. If we need to validate data on client side itself, we have to set a property to controls to execute at the client side.

**3. What is the appSettings Section in the *web.config* file?**

The *web.config* file sets the configuration for a Web project. The *appSettings* block in configuration file sets the user-defined values for the whole application.

For example, in the following code snippet, the specified *ConnectionString* section is used throughout the project for database connection:

*<configuration>  
<appSettings>  
<add key="ConnectionString" value="server=indiabixserver; pwd=dbpassword; database=indiabix" />*

*</appSettings>*

**Skill Practice;**

* Observe the web application page and fill out the format

|  |  |  |
| --- | --- | --- |
| **Control Name Used** | **Control Property Name** | **Control**  **Property Value** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **Methods Used** | | |
|  | | |
|  | | |

**Part B: Ex No: 6 - To Read an XML document containing**

**subjects and marks scored into a dataset**

**Aim:**

To create a window application to read an XML document containing subject, mark scored, year of passing into a Dataset

**Procedure:**

1. Create XML Document with subjects, Marks Scored and Year of Passing.

2. Open a Window Application

3. Design the Form as follows

* Drag and Drop the following controls from the tool box and change the name and text properties.
* 1. Button 2. DataGridView

4. Write the following code in the click () event of the buttons.

5. Save and Execute (press F5).

**Source Code:**

1: using System;

2: using System.Windows.Forms;

3: using System.Xml;

4: using System.Data;

5:

6: namespace Ex\_13

7: {

8: public partial class Form1 : Form

9: {

10: public Form1()

11: {

12: InitializeComponent();

13: }

14: private void rdb\_select\_CheckedChanged()

15: {

16: DataSet DS = new DataSet();

17:

18: DS.ReadXml("E:\\labcode\_cbt\\CBT\_Lab\_N\\Ex\_13

\\student\_mark\_details.xml");

19: dataGridView1.DataSource = DS;

20: dataGridView1.DataMember = "Sem";

21: }

22: }

23: }

**XML (student\_mark\_details.xml) :**

<?xml version="1.0" encoding="utf-8" ?>

<StuDetails>

<Sem>

<sub1>C Programming</sub1>

<Mark1>100</Mark1>

<sub2>BEEE</sub2>

<Mark2>90</Mark2>

<sub3>Operating Systems</sub3>

<Mark3>80</Mark3>

<Year\_of\_passing>2016</Year\_of\_passing>

</Sem>

<Sem>

<sub1>JAVA Programming</sub1>

<Mark1>100</Mark1>

<sub2>Data Structures and Algorithm</sub2>

<Mark2>90</Mark2>

<sub3>Computer Architecture</sub3>

<Mark3>80</Mark3>

<Year\_of\_passing>2016</Year\_of\_passing>

</Sem>

<Sem>

<sub1>.NET Programming</sub1>

<Mark1>100</Mark1>

<sub2>Web Programming</sub2>

<Mark2>90</Mark2>

<sub3>RDBMS</sub3>

<Mark3>80</Mark3>

<Year\_of\_passing>2017</Year\_of\_passing>

</Sem>

</StuDetails>

**Output:** (Print your screen for output)

**Result:**

The windows application was created with datagridview to read xml document containing students mark details and executed to view in the datagridview control.

**Viva Questions:**

1. **What is XSLT?**

XSLT is Extensible Stylesheet Language Transformations that is a part of XML, which is a mechanism to transform an XML document into another XML or HTML document.

1. **What is an attribute?**

An attribute provides more or additional information about an element than otherwise.

Example: <Student name=”raja”> </Student>

1. **What is XML Element?**

An XML document contains XML Elements, and it starts from an element’s start tag to end tag. It can contain:

* + Other elements within main element
  + An Attribute
  + Text

1. **What is XQuery?**

XQuery was designed to query XML data which is nothing but SQL for database tables. XQuery is used to fetch the data from the XML file.

1. **What is XML Encoding?**

XML documents may contain Non-ASCII characters like French and Norwegian characters. XML Encoding is used to avoid errors and XML files have to be saved as Unicode.

**Skill Practice;**

* Create a xml file for employee HR- Payroll salary calculation

**Part B: Ex No: 7 - To Read Students Record from Database using ADO.Net and Store in XML Document**

**Aim:**

To Develop a Window application to read students records from Database using ADO.NET and generate XML document containing students records

**Procedure:**

1. Create a Database and a table to store Students Records.

2. Open a Window Application

3. Design the Form to insert Students Details

* Drag and Drop the following controls from the tool box and change the name and text properties.

1. Button 2. DataGridView 3. Label Box 4.DropDownbox

5. TextBox

4. Write the following code in the click () event of the buttons.

5. Save and Execute (press F5).

**Source Code:**

1: using System;

2: using System.Data;

3: using System.Data.SqlClient;

4: using System.Windows.Forms;

5:

6: namespace Ex\_14

7: {

8: public partial class Form1 : Form

9: {

10: public Form1()

11: {

12: InitializeComponent();

13: }

14: //CONNECTION OBJECT

15: SqlConnection con = new SqlConnection();

16: // COMMAND OBJECT

17: SqlCommand cmd = new SqlCommand();

18: DataSet ds = new DataSet();

19: private void rdb\_select\_CheckedChanged()

20: {

21: openFileDialog1.ShowDialog();

22: con.ConnectionString = @"Data

Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=" +

openFileDialog1.FileName + @";Integrated

Security=True;Integrated Security=True";

23: con.Open();

24: cmd.Connection = con;

25: cmd.CommandText = "select \* from student\_master ";

26: SqlDataAdapter ada = new SqlDataAdapter();

27: ada.SelectCommand = cmd;

28: ada.Fill(ds, "student\_master");

29: dataGridView1.DataSource = ds;

30: dataGridView1.DataMember = "student\_master";

31: }

32: private void btn\_save\_Click(object sender, EventArgs e)

33: {

34: ds.WriteXml("E:\\labcode\_cbt\\CBT\_Lab\_N

[\\Ex\_14\\student\_details.xml](file:///\\Ex_14\\student_details.xml)", XmlWriteMode.WriteSchema);

35: con.Close();

36: MessageBox.Show("Sucessfully Written students data

from database to XML...");

37: }

38:

39: private void btnInsert\_Click(object sender, EventArgs e)

40: {

41: int x;

42: SqlConnection con = new SqlConnection();

**43: // Define database connection string**

44: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

45: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

46: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

47: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

48: con.Open();

49: SqlCommand cmd = new SqlCommand();

50: cmd.Connection = con;

51: // Define the sql commenad to Insert students records

52: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

53: x = cmd.ExecuteNonQuery();

54: if (x == 1)

55: MessageBox.Show("Students Details Saved");

56: else

57: MessageBox.Show("Not Saved");

58: }

59:

60: }

61: }

**Output:** (Print the following outputs)

1. ADO.NET Execution
2. XML File Written data from database

**Result:**

The application was created with ADO.Net to store student’s records and the data were written to the xml document in the specified directory.

**Viva Questions and Answers:**

1. **Why Is Xml Such An Important Development?**

It removes two constraints which were holding back Web developments:

1. Dependence on a single, inflexible document type (HTML) which was being much abused for tasks it was never designed for;

2. The complexity of full SGML, whose syntax allows many powerful but hard-to-program options. XML allows the flexible development of user-defined document types.

2. **What Is SGML?**

SGML is the Standard Generalized Markup Language (ISO 8879:1986), the international standard for defining descriptions of the structure of different types of electronic document.

1. **How Do I Use Graphics In Xml?**

NOTATION and ENTITY