Working with Data (Form Control + Database)

Content

Form Controls

1. **DataGridView**
2. **ListView**

1. DataGridView Control

DataGridView control is designed to be a complete solution for displaying tabular data with Windows Forms. This control makes it easy to define the basic appearance of cells and the display formatting of cell values.

The **Cell** is the fundamental unit of interaction for the DataGridView. All cells derive from the DataGridViewCell base class. Each cell within the DataGridView control can have its own style, such as text format, background color, foreground color, and font.

The DataGridView control is highly configurable and extensible, and it provides many properties, methods, and events to customize its appearance and behavior.

DataGridView Properties:

- Columns property
- Rows property

More Properties at *https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.datagridview*

Runtime Examples:

Setup DataGridView (Control name: dgvProfile)

```
dgvProfile.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
dgvProfile.MultiSelect = false;
dgvProfile.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;
```

Add Columns

```
dgvProfile.ColumnCount = 3;
dgvProfile.Columns[0].Name = "Column 1";
dgvProfile.Columns[1].Name = "Column 2";
dgvProfile.Columns[2].Name = "Column 3";
```

Change Header

```
string[] header = { "First Name", "Middle Name", "Last Name" };
foreach (DataGridViewColumn dgvColumn in dgvProfile.Columns)
{
        dgvColumn.HeaderText = header[dgvColumn.Index];
}
```

Add Row

```
string[] name = { txtFname.Text, txtMiddle.Text, txtLast.Text };
dgvProfile.Rows.Add(name);
```

Add button to DataGridView

```
DataGridViewButtonColumn btnDGV = new DataGridViewButtonColumn();
dgvProfile.Columns.Add(btnDGV);
btnDGV.HeaderText = "Action";
btnDGV.Text = "Delete";
btnDGV.UseColumnTextForButtonValue = true;
```

DataGridView CellClick Event

```
if(e.ColumnIndex == 3)
{
        if (dgvProfile.SelectedRows.Count > 0)
        {
                dgvProfile.Rows.RemoveAt(dgvProfile.Rows[e.RowIndex].Index);
        }
}
```

DataGridView cellContentClick Event

```
if(dgvProfile.SelectedRows.Count > 0)
{
        DataGridViewRow row = dgvProfile.SelectedRows[0];
        txtFname.Text = row.Cells[0].Value + string.Empty;
        txtMiddle.Text = row.Cells[1].Value + string.Empty;
        txtLast.Text = row.Cells[2].Value + string.Empty;
}
```

Backcolor of DataGridView Cell

```
dgvProfile.Rows[0].Cells[2].Style.BackColor = Color.Red;
```

2. ListView Control

C# ListView control provides an interface to display a list of items using different views including text, small images, and large images.

There are two approaches to create a ListView control in Windows Forms. Either we can use the Forms designer to create a control at design-time or we can use the ListView class to create a control at run-time.

ListView Methods, Properties, & Events

- *Columns.Add() method.* You can add columns in Listview by using Columns.Add() method.
- *ListViewItem*. You can add items in listbox using ListViewItem which represents an item in a ListView control.
- *Sorted property*. If the Sorted property of Listview is set to true, then the ListView items are sorted.

More Properties at *https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.listview*

Runtime Example:

Setup ListView Control (Control Name: lvProfile)

```
lvProfile.BorderStyle = BorderStyle.FixedSingle;
lvProfile.FullRowSelect = true;
lvProfile.GridLines = true;
lvProfile.View = View.Details;
lvProfile.MultiSelect = false;
```

Add Column

```
lvProfile.Columns.Add("Column 0");
lvProfile.Columns.Add("Column 1");
lvProfile.Columns.Add("Column 2");
```

Modify ListView Header

```
lvProfile.Columns[0].Text = "First Name";
lvProfile.Columns[0].Width = 100;
lvProfile.Columns[1].Text = "Middle Name";
lvProfile.Columns[1].Width = 100;
lvProfile.Columns[2].Text = "Last Name";
lvProfile.Columns[2].Width = 100;
```

Add ListView item

```
ListViewItem lvi = new ListViewItem(txtFname.Text);
lvi.SubItems.Add(txtMiddle.Text);
lvi.SubItems.Add(txtLast.Text);
lvProfile.Items.Add(lvi);
```

ListView SelectedIndexChanged Events

```
if (lvTable.SelectedItems.Count == 0)
        return;
ListViewItem item = lvTable.SelectedItems[0];
txtFirst.Text = item.SubItems[0].Text;
txtMiddle.Text = item.SubItems[1].Text;
txtLast.Text = item.SubItems[2].Text;
```

References

https://www.hostinger.com/tutorials/what-is-mysql

https://dev.mysql.com/doc/dev/connector-net/6.10/html/T_MySql_Data_MySqlClient_MySqlCommand.htm

https://www.delftstack.com/howto/csharp/mysql-connection-in-csharp/

https://dev.mysql.com/doc/dev/connector-net/6.10/html/Overload_MySql_Data_MySqlClient_MySqlDataAdapter__ctor.htm

http://csharp.net-informations.com/datagridview/csharp-datagridview-tutorial.htm

https://www.c-sharpcorner.com/UploadFile/mahesh/working-with-listview-in-C-Sharp/#:~:text=The%20ListView%20in%20C%23%20provides,small%20images%2C%20and%20large%20images.

http://csharp.net-informations.com/gui/cs-listview.htm

Rubrics

Output	Excellent(4)	Good(3)	Fair(2)	Poor (1-0)
Program Execution	Program executes correctly with no syntax or runtime errors	Program executes with a minor error (easily fixed error)	Program executes with many errors	Program does not execute
Correct Output	Program displays correct output with no errors	Output has minor errors	Output has multiple errors	Output is incorrect
Logic	Program is created logically well	Program has slight logic errors that do no significantly affect the results	Program has significant logic errors	Program is incorrect
Code Hygiene & Readability	Code is clean, understandable, well-organized	Minor issues such as inconsistent indentation, variable naming, general organization	At least one major issue that makes it difficult to read	Several major issues that make it difficult to read.
Documentation	Code is well commented.	One or two places could benefit from	Major lack of comments	No comments.

		comments, or the code is overly commented	make it difficult to understand code.	
Timeliness / Delivery	The program was submitted on time	The program was submitted within three days of the due date.	The program was submitted within four to seven days of the due date.	The program was submitted more than seven days overdue.