**Masked textbox**

A Masked Textbox control provides validation mechanism for user input on a Form. For example, if you want a Textbox to accept date in mm/dd/yyyy format, you can set masking in the MaskedTextBox.

0 - Digit, required. Value between 0 and 9.

9 - Digit or space, optional.

# - Digit or space, optional. If this position is blank in the mask, it will be rendered as a space in the Text property.

L - Letter, required. Restricts input to the ASCII letters a-z and A-Z.

? - Letter, optional. Restricts input to the ASCII letters a-z and A-Z.

& - Character, required.

C - Character, optional. Any non-control character.

A - Alphanumeric, required.

a - Alphanumeric, optional.

.  - Decimal placeholder.

, - Thousands placeholder.

: - Time separator.

/ - Date separator.

$ - Currency symbol.

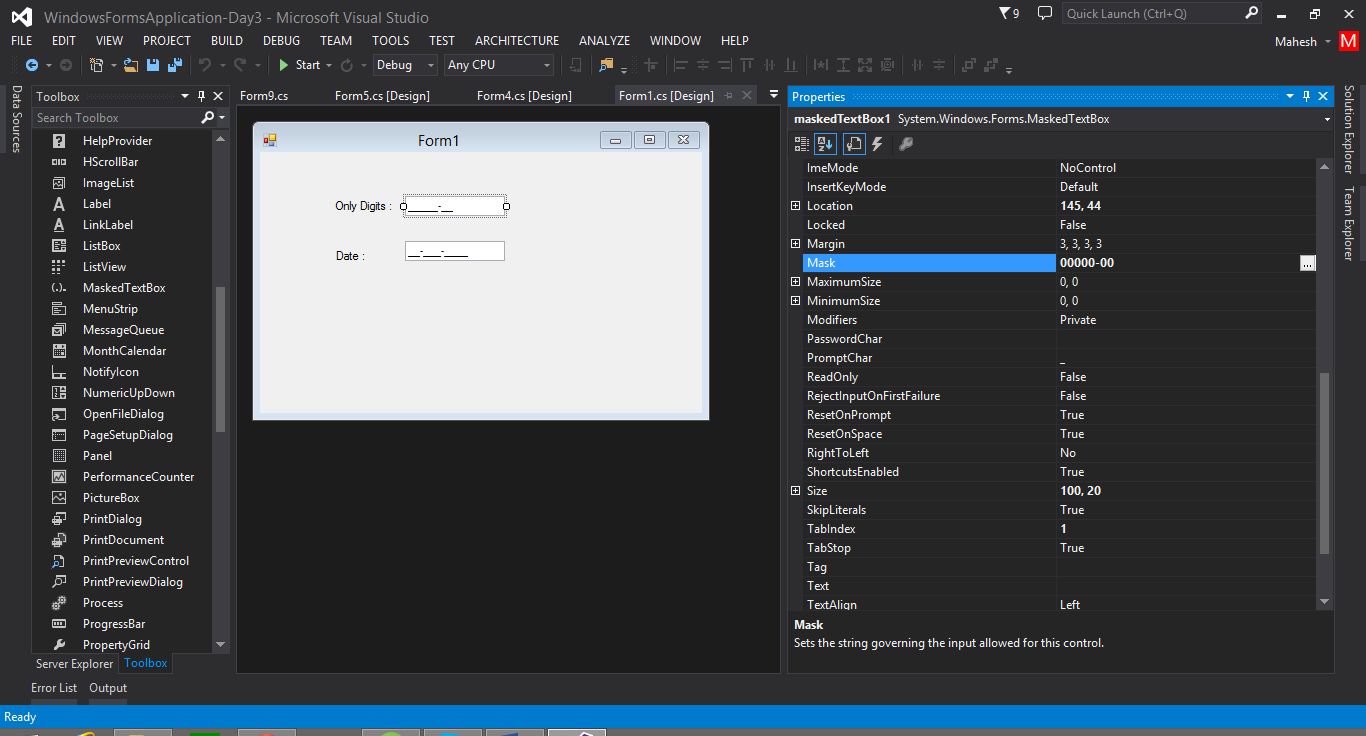
< - Shift down. Converts all characters that follow to lowercase.

> - Shift up. Converts all characters that follow to uppercase.

| - Disable a previous shift up or shift down.

\ - Escape. Escapes a mask character, turning it into a literal. "\\" is the escape sequence for a backslash.

All other characters - Literals. All non-mask elements will appear as themselves within Masked Textbox. Literals always occupy a static position in the mask at run time, and cannot be moved or deleted by the user.



1. Select masked textbox goto properties set mask=00000-00 (only for digits)
2. Mast=LLLLL (only for letters)

**Timer**

The **Timer** control allows you to set a time interval to periodically execute an event at a specified interval. It is useful when you want to execute certain applications after a certain interval

Constructor: timer

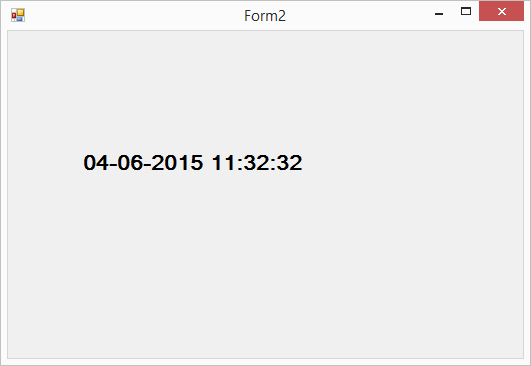
Properties: Enabled, interval

Events: tick

Methods: start (), stop ()

1 sec = 1000 milliseconds.

1. Digital clock



namespace WindowsFormsApplication\_Day3

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

private void Form2\_Load(object sender, EventArgs e)

{

timer1.Start();

}

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

{

label1.Text = DateTime.Now.ToString();

}

}

}

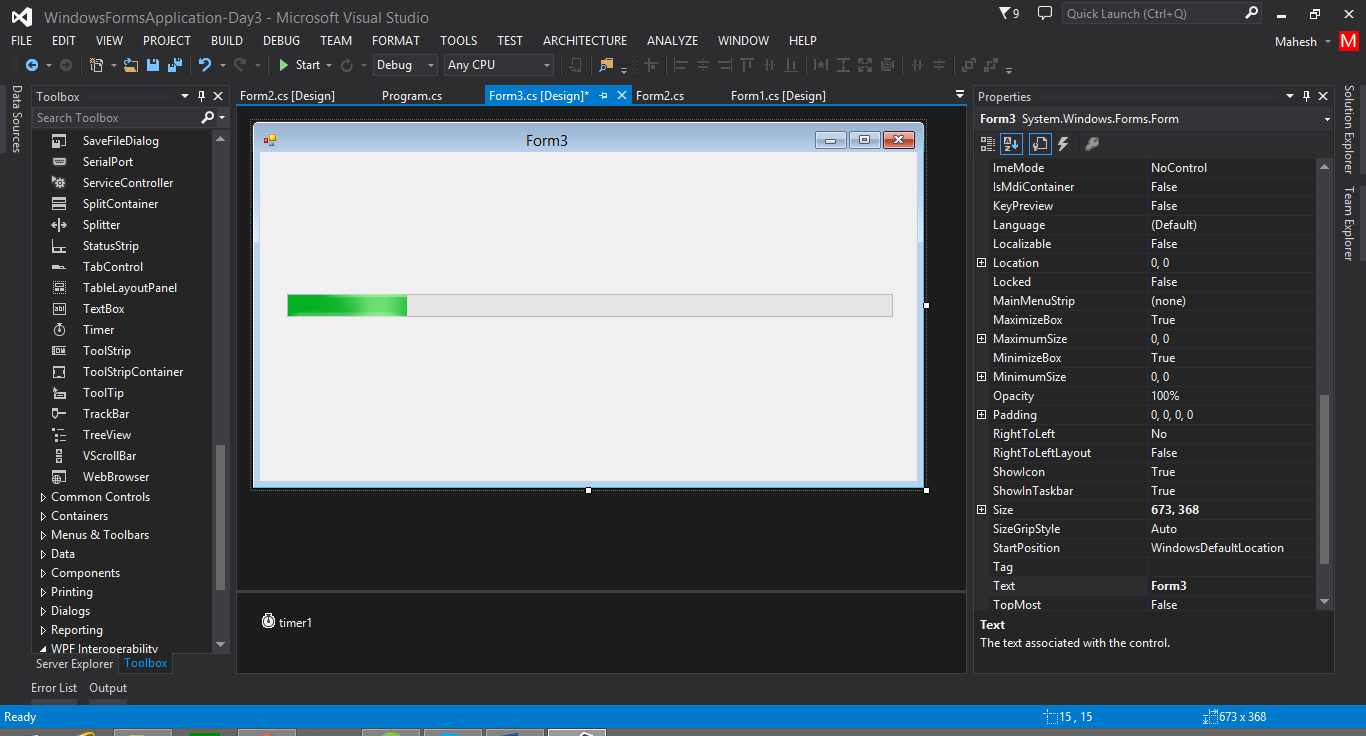
**Progress bar**

A progress bar is a control that an application can use to indicate the progress of a lengthy operation such as calculating a complex result, downloading a large file from the Web etc. Progress Bar controls are used whenever an operation takes more than a short period of time. The Maximum and Minimum properties define the range of values to represent the progress of a task.

Constructor: Progress Bar

Properties: Maximum, minimum,value

Program: 1



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.Windows.Forms;

namespace WindowsFormsApplication\_Day3

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

private void Form3\_Load(object sender, EventArgs e)

{

// start timer

timer1.Start();

}

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

{

if(progressBar1.Value>=progressBar1.Maximum)

{

timer1.Stop();

MessageBox.Show(" Your Prograss is Completed...");

}

else

{

progressBar1.Value = progressBar1.Value + 10;

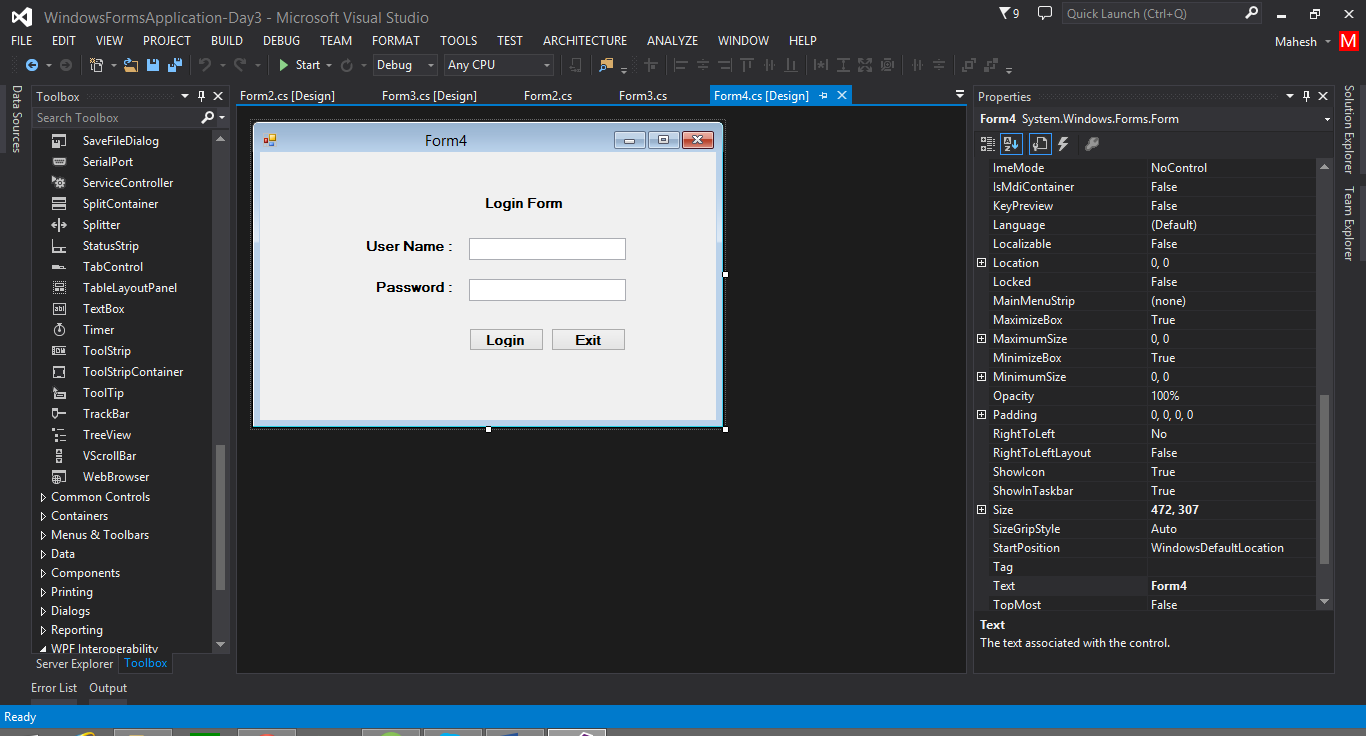
}

}

}

}

Program 2



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.Windows.Forms;

namespace WindowsFormsApplication\_Day3

{

public partial class Form4 : Form

{

public Form4()

{

InitializeComponent();

}

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

{

if(textBox1.Text=="admin" && textBox2.Text=="admin")

{

Form5 obj = new Form5();

obj.Show();

this.Hide();

}

else

{

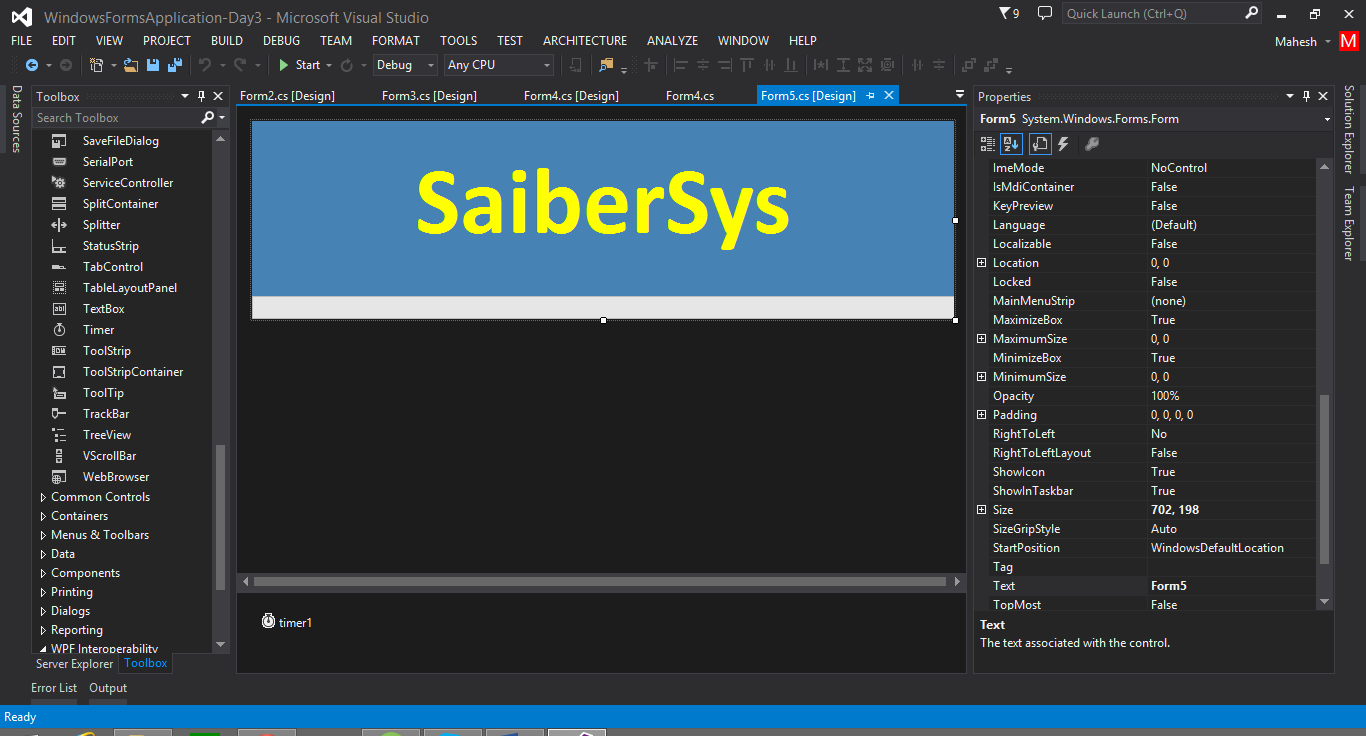
MessageBox.Show(" Invalid user name and password....");

}

}

}

}



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.Windows.Forms;

namespace WindowsFormsApplication\_Day3

{

public partial class Form5 : Form

{

public Form5()

{

InitializeComponent();

}

private void Form5\_Load(object sender, EventArgs e)

{

timer1.Start();

}

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

{

if (progressBar1.Value >= progressBar1.Maximum)

{

timer1.Stop();

progressBar1.Visible = false;

}

else

{

progressBar1.Value = progressBar1.Value + 10;

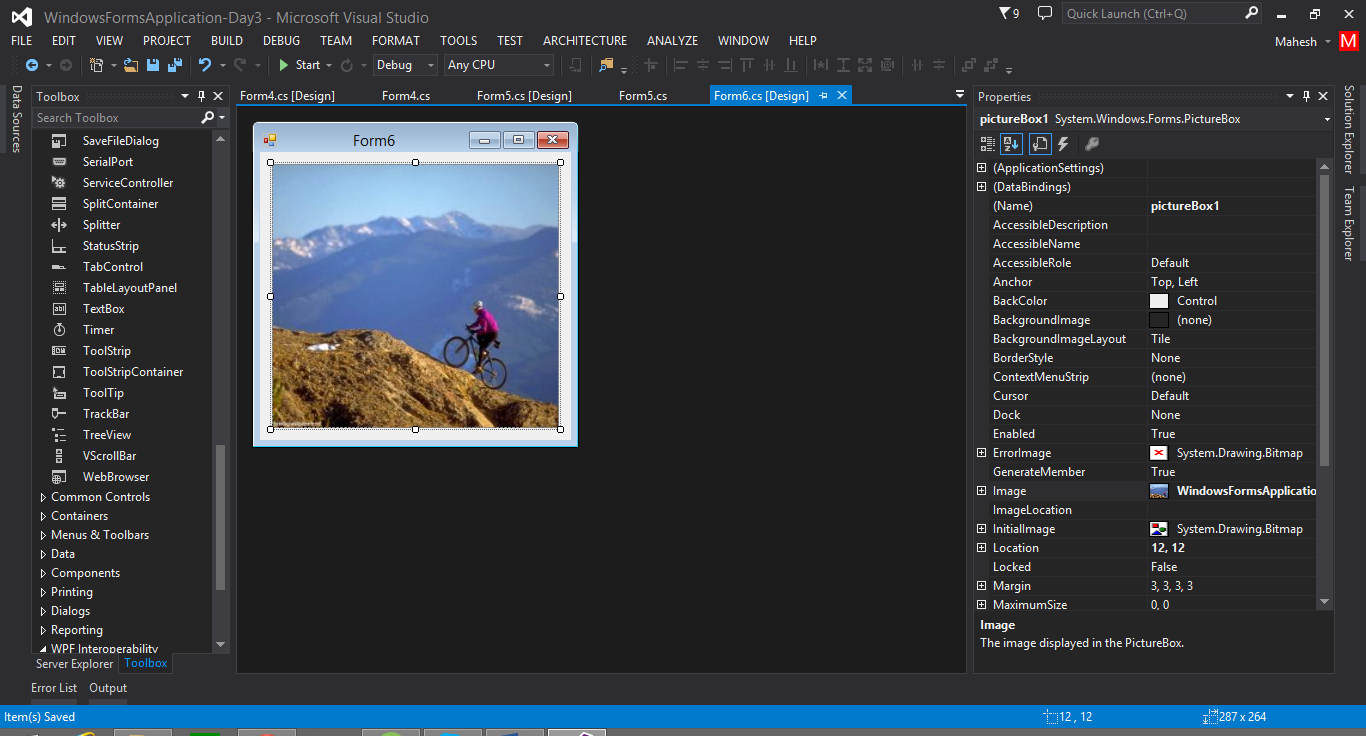
}

}

}

}

**Picture box**

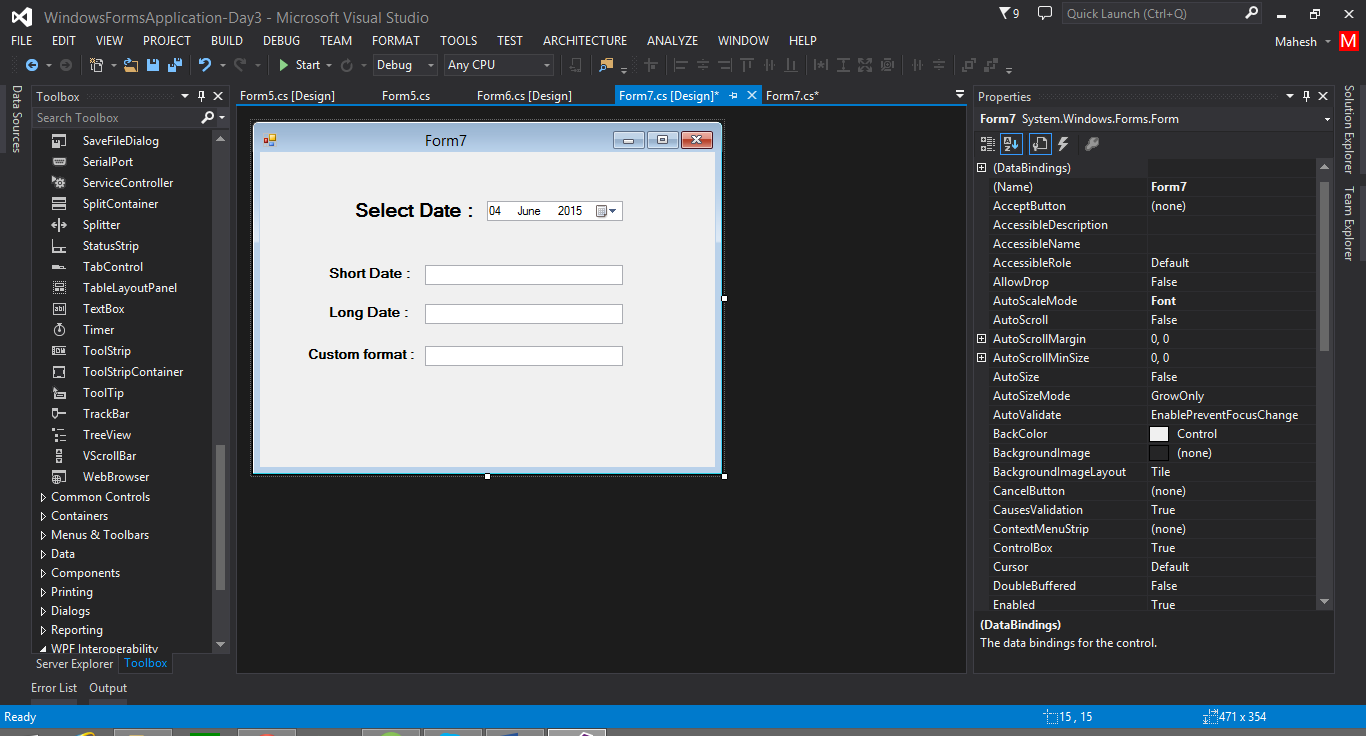


Properties :

Image= (import image Project/resource )

Sizemode=StretchImage

**Date time picker**



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.Windows.Forms;

namespace WindowsFormsApplication\_Day3

{

public partial class Form7 : Form

{

public Form7()

{

InitializeComponent();

}

private void dateTimePicker1\_ValueChanged(object sender, EventArgs e)

{

textBox1.Text = dateTimePicker1.Value.ToShortDateString();

textBox2.Text = dateTimePicker1.Value.ToLongDateString();

textBox3.Text = dateTimePicker1.Value.ToString("dd / MM / yyyy hh:mm:ss");

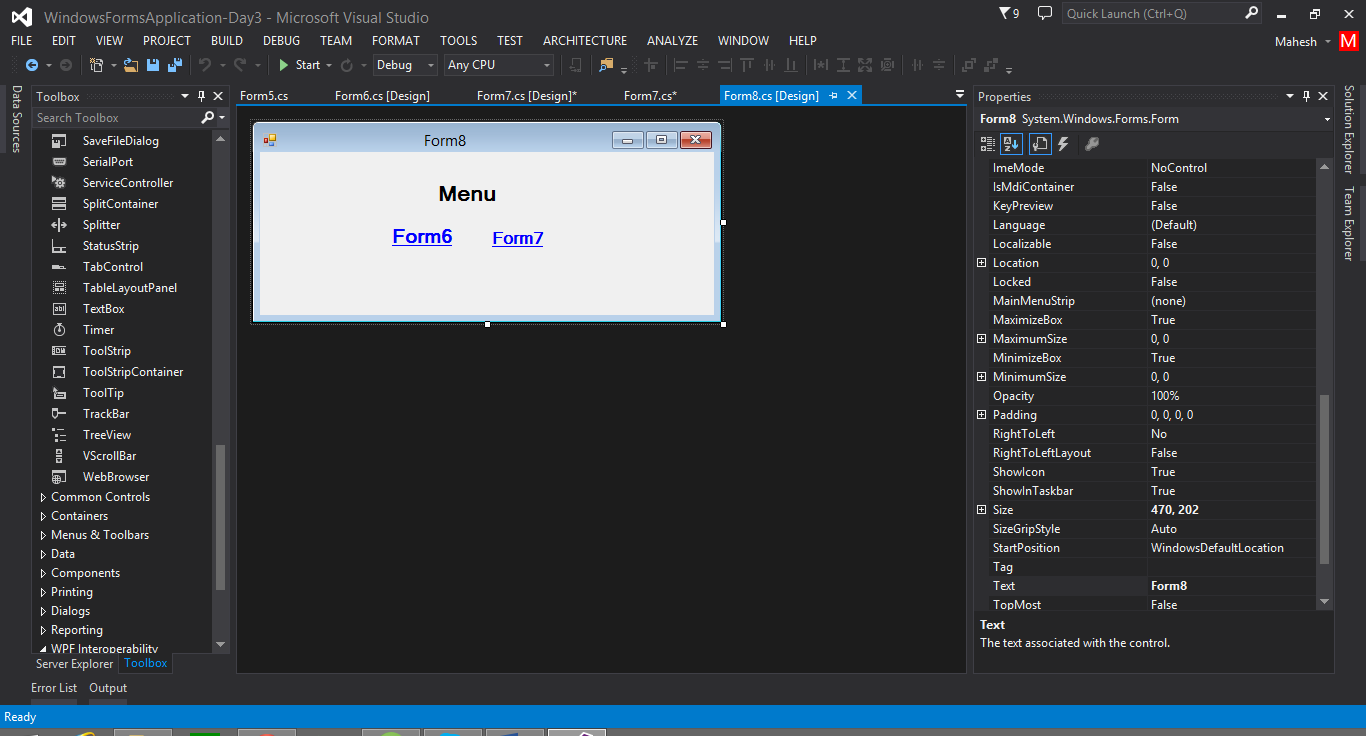
}

}

}

**Link label:** A Link Label control is a label control that can display a hyperlink

Program: using Link label open the another form



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.Windows.Forms;

namespace WindowsFormsApplication\_Day3

{

public partial class Form8 : Form

{

public Form8()

{

InitializeComponent();

}

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

//open form 6

Form6 obj = new Form6();

obj.Show();

}

private void linkLabel2\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

// open form7

Form7 obj = new Form7();

obj.Show();

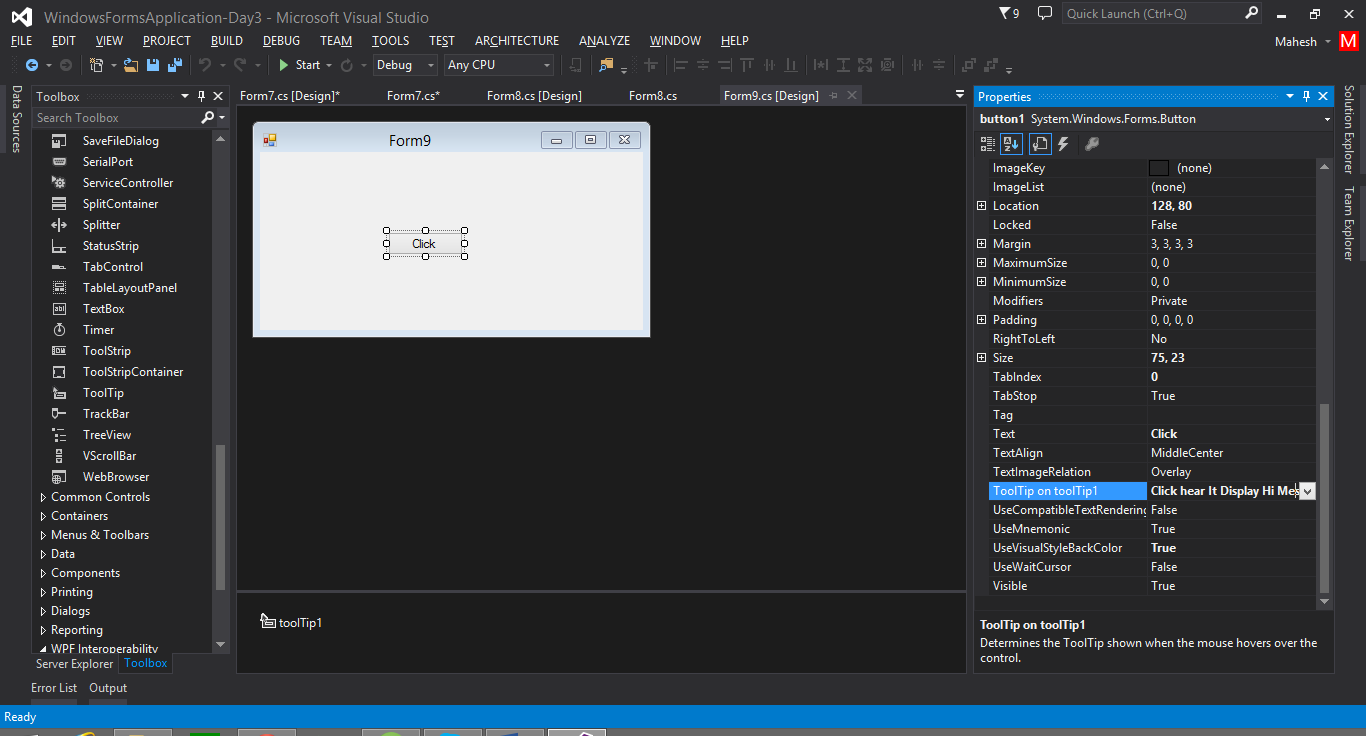
}

}

}

**Tool Tip:**

A tooltip is a small pop-up window that displays some information when you rollover on a control.



1. Select tooltip from the toolbox and place in form
2. Select any control in your form and got o their properties and write your control information in tooltip