VB.NET MDI Form

MDI stands for **Multiple Document Interface** applications that allow users to work with multiple documents by opening more than one document at a time. Whereas, a **Single Document Interface (SDI)** application can manipulate only one document at a time.

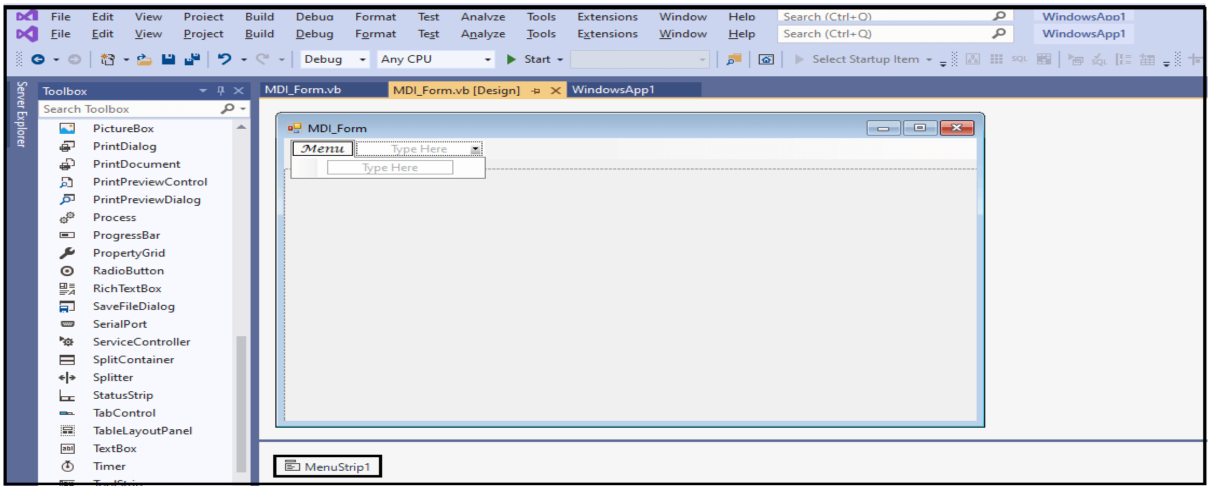
The MDI applications act as the parent and child relationship in a form. A parent form is a container that contains child forms, while child forms can be multiple to display different modules in a parent form.

VB.NET has following rules for creating a form as an MDI form.

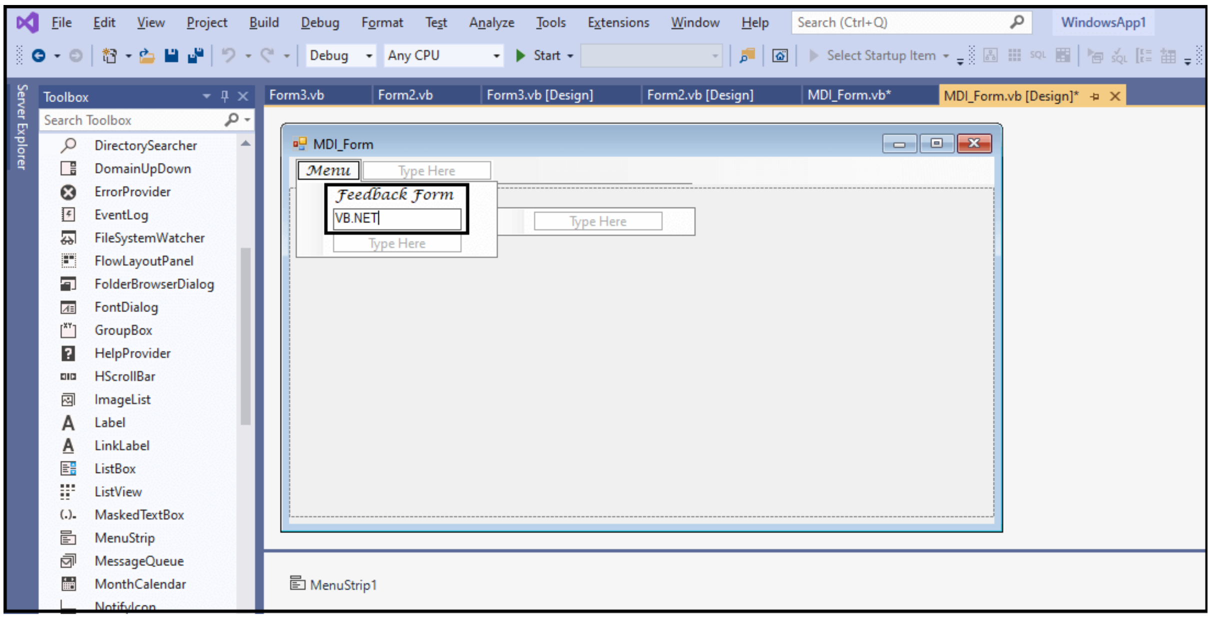
1. **MidParent:** The MidParent property is used to set a parent form to a child form.
2. **ActiveMdiChild:** The ActiveMdiChild property is used to get the reference of the current child form.
3. **IsMdiContainer:** The IsMdiContainer property set a Boolean value to True that represents the creation of a form as an MDI form.
4. **LayoutMdi():** The LayoutMdi() method is used to arrange the child forms in the parent or main form.
5. **Controls:** It is used to get the reference of control from the child form.

Let's create a program to display the multiple windows in the VB.NET Windows Forms.

Step 1: First, we have to open the Windows form and create the Menu bar with the use of MenuStrip control, as shown below.



Step 2: After creating the Menu, add the Subitems into the Menu bar, as shown below.



In the above image, we have defined two Subitems, First is the **Feedback Form**, and the Second is VB.NET.

Step 3: In the third step, we will create two Forms: The Child Form of the **Main Form** or **Parent Form**.

Here, we have created the first Child Form with the name Form2.

Form2.vb

```
Public Class Form2
    Private Sub Form2_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Me.Text = "Feedback form" ' Set the title of the form
        Label1.Text = " Fill the Feedback form"
        Button1.Text = "Submit"
        Button1.BackColor = Color.SkyBlue
        Button2.Text = "Cancel"
        Button2.BackColor = Color.Red
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

        MsgBox(" Successfully submit the feedback form")
    End Sub
    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
         Me.Dispose() ' end the form2
    End Sub
End Class
```

Another **Child Form** with the name **Form3**.

Form3.vb

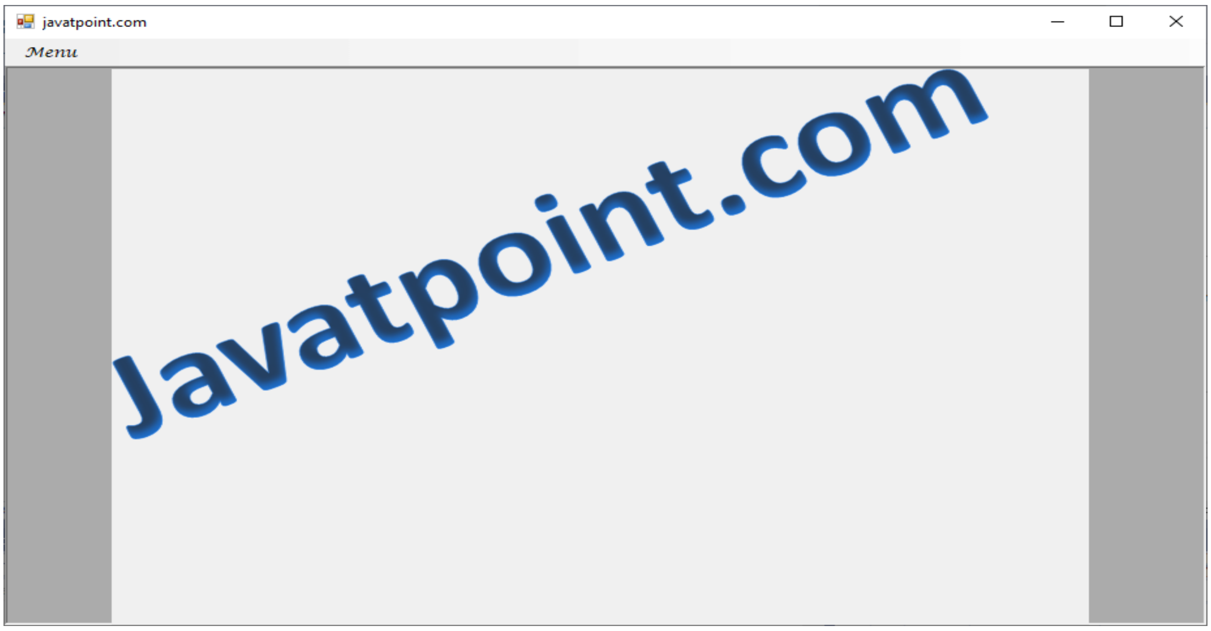
```
Public Class Form3
    Private Sub Form3_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Label1.Text = "Welcome to JavaTpoint Tutorial Site"
        Label.BackColor = Color.Green
Label2.Text = "This is the VB.NET Tutorial and we are learning the VB.NET MDI Form"

        Label2.BackColor = Color.SkyBlue
    End Sub
End Class
```

Step 4: Now we write the programming code for the Main or Parent Form, and here is the code for our Main Form.

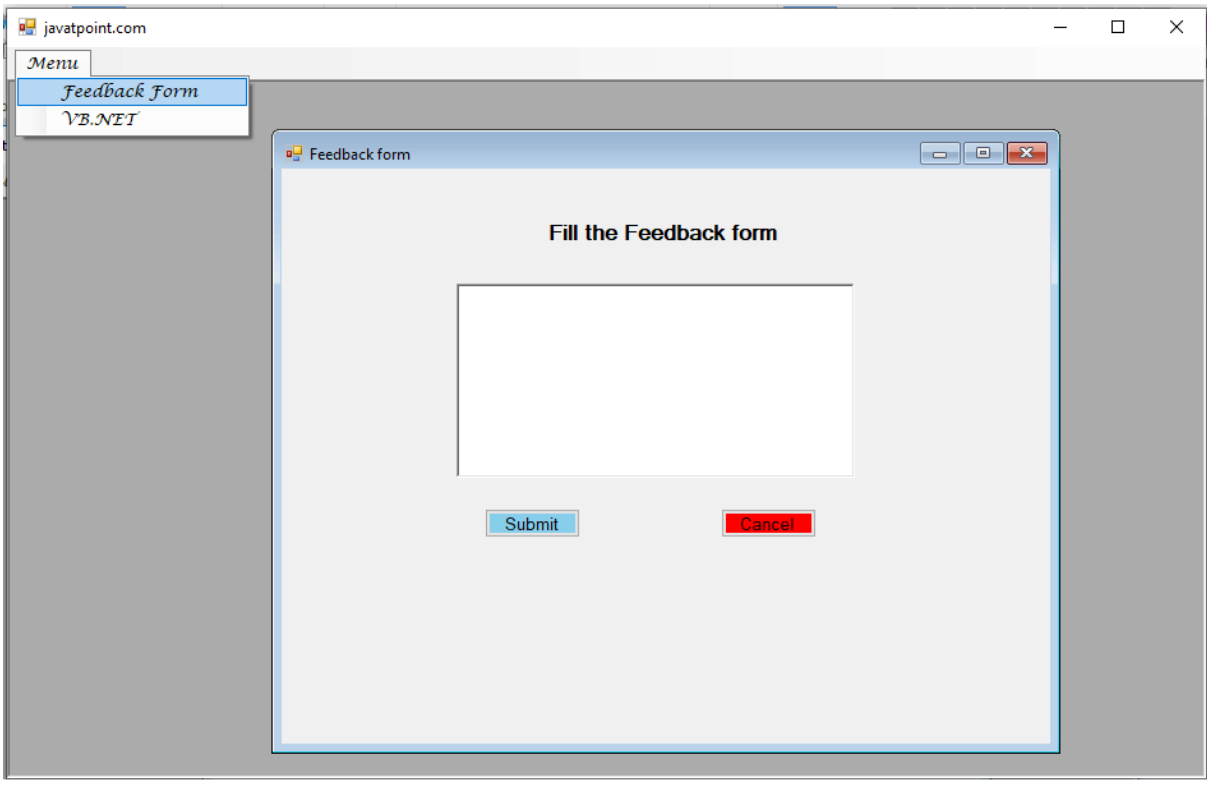
MDI_form.vb

```
Public Class MDI_Form
    Private Sub MDI_Form_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        IsMdiContainer = True 'Set the Boolean value to true to create the form as an MDI form
.       Me.Text = "javatpoint.com" 'set the title of the form
        PictureBox1.Image = Image.FromFile("C:\Users\AMIT YADAV\Desktop\jtp2.png")
        PictureBox1.Height = 550
        PictureBox1.Width = 750
    End Sub
    Private Sub FeedbackFormToolStripMenuItem_Click(sender As Object, e As EventArgs) Ha
ndles FeedbackFormToolStripMenuItem.Click
        PictureBox1.Visible = False
        Dim fm2 As New Form2
        fm2.MdiParent = Me 'define the parent of form3, where Me represents the same form
        fm2.Show() 'Display the form3
    End Sub
    Private Sub VBNETToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles VB
NETToolStripMenuItem.Click
        PictureBox1.Visible = False
        Dim fm3 As New Form3
        fm3.MdiParent = Me 'define the parent of form3, where Me represent the same form
        fm3.Show() 'Display the form3
    End Sub
End Class
```

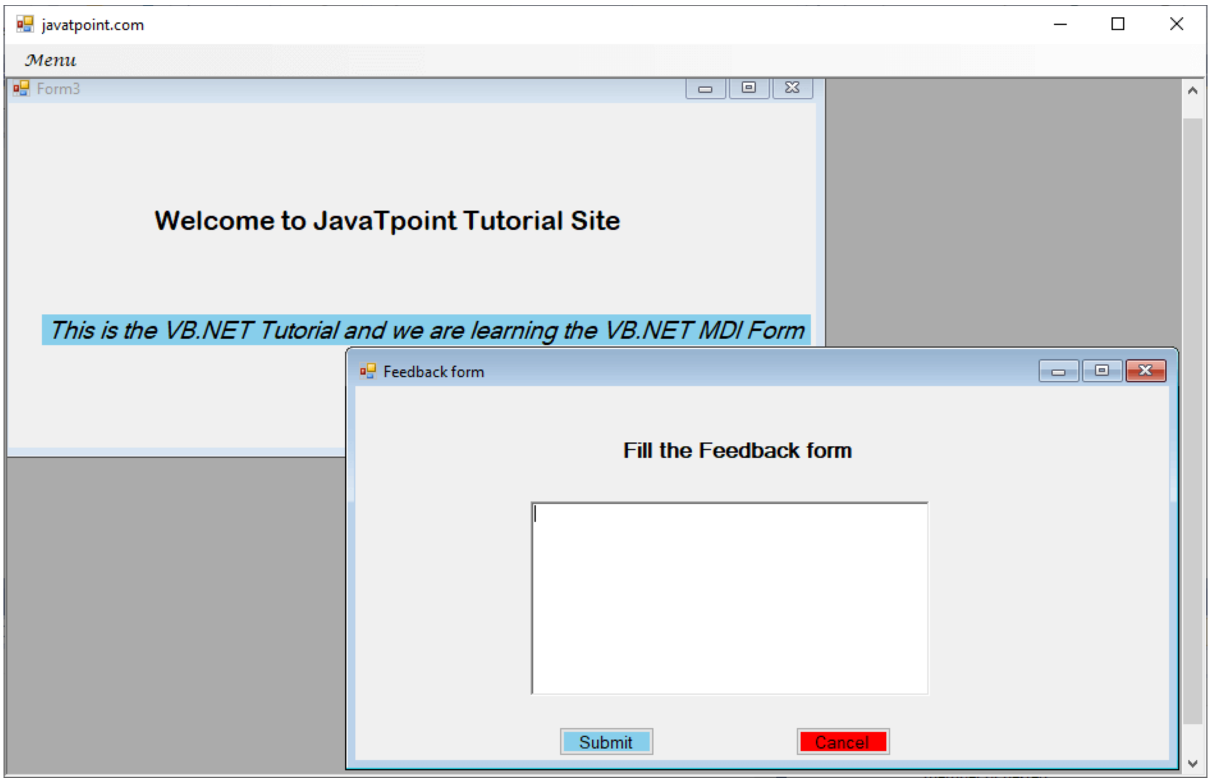
Output:



After that, click on the Menu button, it shows two sub-items of the Menu as **Feedback Form** and **VB.NET**. We have clicked on the Feedback Form that displays the following form on the window.



When we click on the Menu item, it shows the following image on the screen.

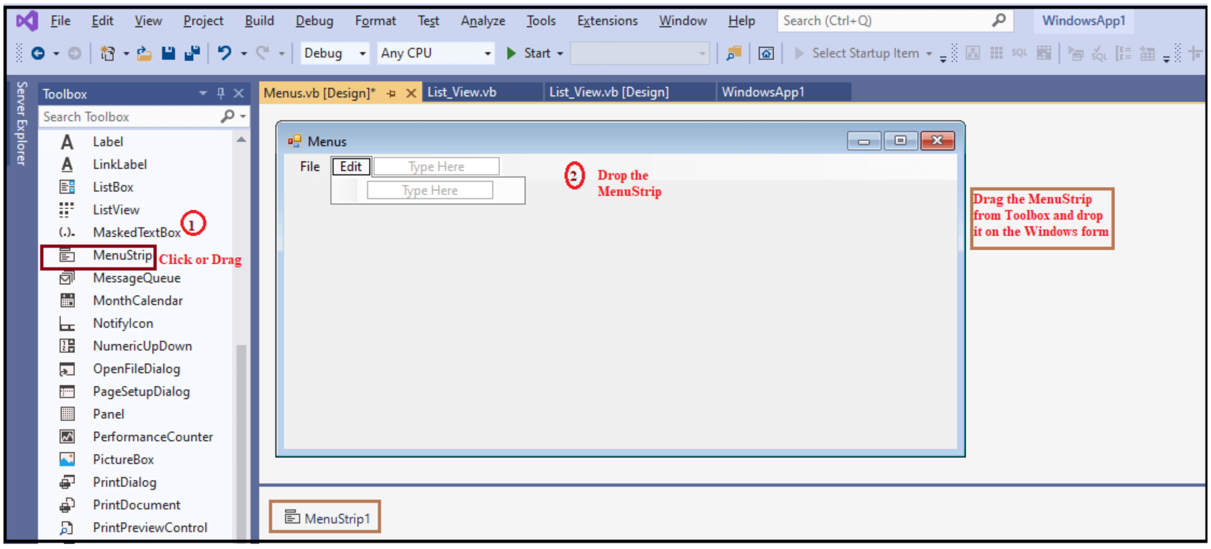

javatpoint.com
Menu
Form3
Welcome to JavaTpoint Tutorial Site
This is the VB.NET Tutorial and we are learning the VB.NET MDI Form
Feedback form
Fill the Feedback form
Submit
Cancel

VB.NET Menu Control

A menu is used as a menu bar in the Windows form that contains a list of related commands, and it is implemented through MenuStrip Control. The Menu control is also known as the VB.NET **MenuStrip** Control. The menu items are created with ToolStripMenuItem Objects. Furthermore, the **ToolStripDropDownMenu** and **ToolStripMenuItem** objects enable full control over the structure, appearance, functionalities to create menu items, submenus, and drop-down menus in a VB.NET application.

Let's create a MenuBar by dragging a MenuStrip control from the toolbox and dropping it to the Windows form.

Step 1. Drag the MenuStrip control from the toolbox and drop it on to the Form.



Step 2: Once the MenuStrip is added to the form, we can set various properties of the Menu by clicking on the MenuStrip control.

Properties of the MenuStrip Control

There are following properties of the VB.NET MenuStrip control.

Properties	Description

CanOverflow	The CanOverflow property is used to authenticate whether the control supports overflow functionality by setting values in the MenuStrip control.
Stretch	The Stretch property is used to obtain a value that specifies whether the menustrip stretches from end to end in the MenuStrip control.
GripStyle	The GripStyle property obtains or sets the visibility of the grip that uses the reposition of the menu strip control.
ShowItemToolTips	It is used to obtain or set the value that determines if the ToolTips are displayed for the MenuStrip Control.
DefaultSize	The DefaultSize property is used to get the default horizontal and vertical dimension of the MenuStrip in pixel when it is first created.

Methods of the MenuStrip Control

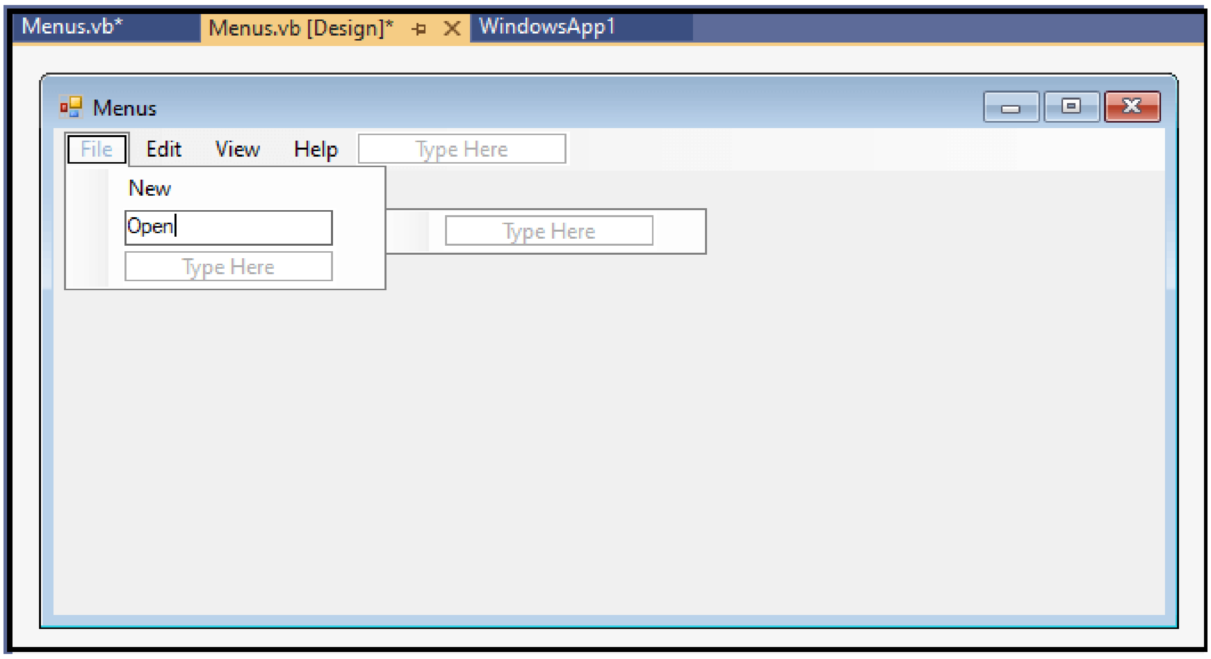
Method	Description
CreateAccessibilityInstance()	It is used to create a new accessibility instance for the MenuStrip Control.
ProcessCmdKey()	The ProcessCmdKey method is used to process the command key in the MenuStrip Control.
CreateDefaultItem()	The CreateDefaultItem method is used to create a ToolStripMenuItem with the specified text, image, and event handlers for the new MenuStrip.
OnMenuActivate()	It is used to initiate the MenuActivate event in the MenuStrip control.
OnMenuDeactivate()	It is used to start the MenuDeactivate event in the MenuStrip control.

Events of the MenuStrip Control

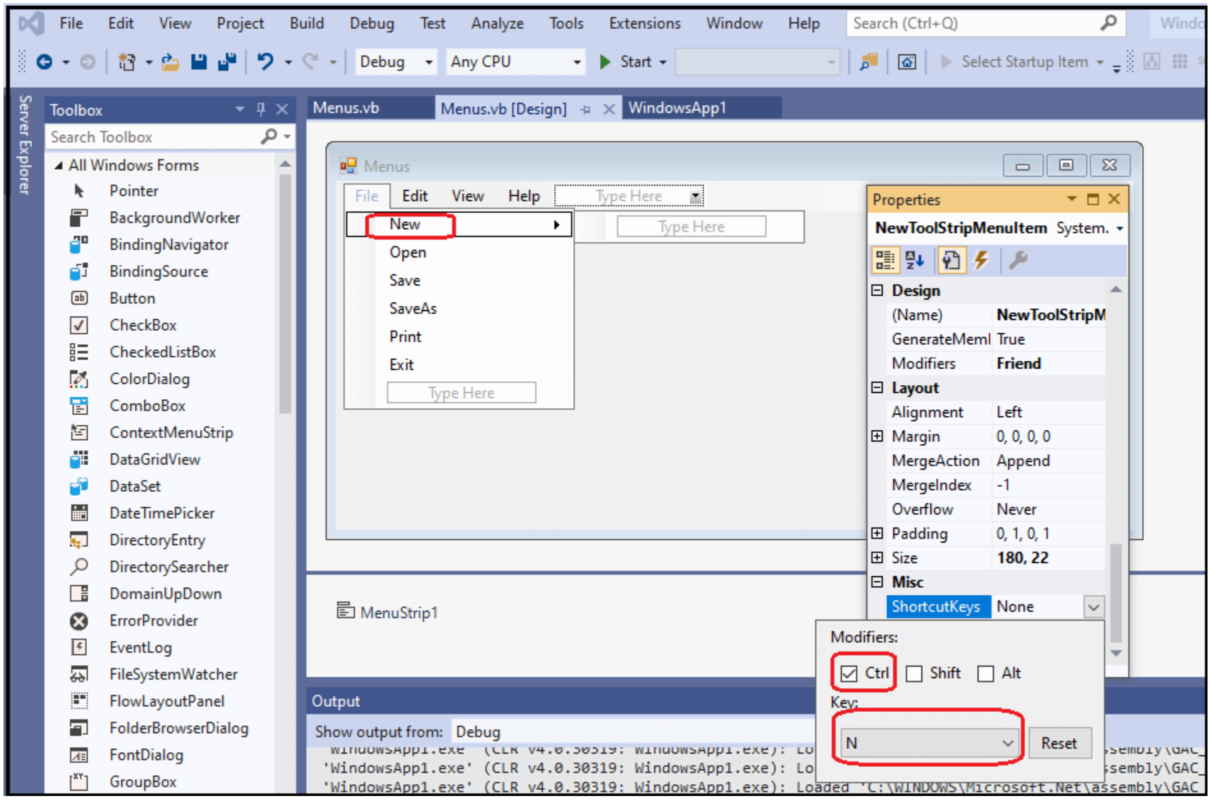
Events	Description
MenuActivate	When a user uses a menu bar control with a mouse or keyboard, a MenuActivate event occurs.
MenuDeactivate	The MenuDeactivate event occurs when the MenuStrip control is deactivated in the Windows form.

Let's create a program to display the menu bar in the Windows form.

In this image, we have created the menu and sub-items of the menu bar in the form.



Now, we write the Shortcut keys for the File subitems, such as **New -> Ctrl + N, Open -> Ctrl + O**, etc.



Properties
NewToolStripMenuItem System.

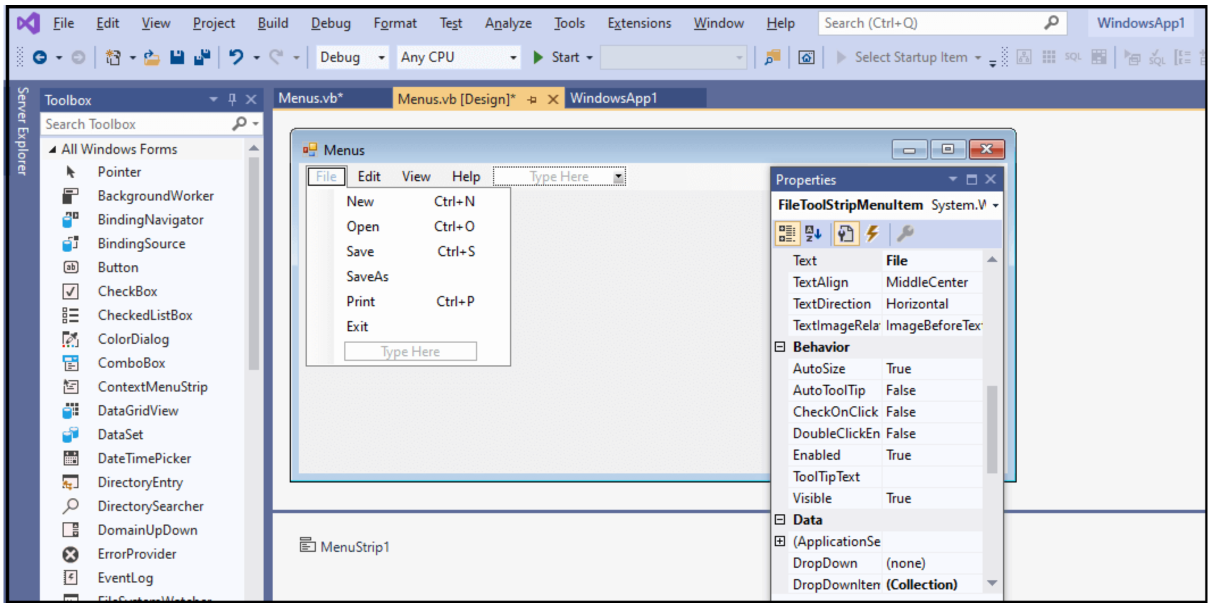
Design	
(Name)	NewToolStripM
GenerateMem	True
Modifiers	Friend
Layout	
Alignment	Left
Margin	0, 0, 0, 0
MergeAction	Append
MergeIndex	-1
Overflow	Never
Padding	0, 1, 0, 1
Size	180, 22
Misc	
ShortcutKeys	None

Modifiers: [✓ Ctrl] Shift Alt
Key: N
Reset

Show output from: Debug

```
'WindowsApp1.exe' (CLR v4.0.30319: WindowsApp1.exe): Lo
'WindowsApp1.exe' (CLR v4.0.30319: WindowsApp1.exe): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC
```

After that, we can see the subitems of the Files with their Shortcut keys, as shown below.



New	Ctrl+N
Open	Ctrl+O
Save	Ctrl+S
SaveAs	
Print	Ctrl+P
Exit	

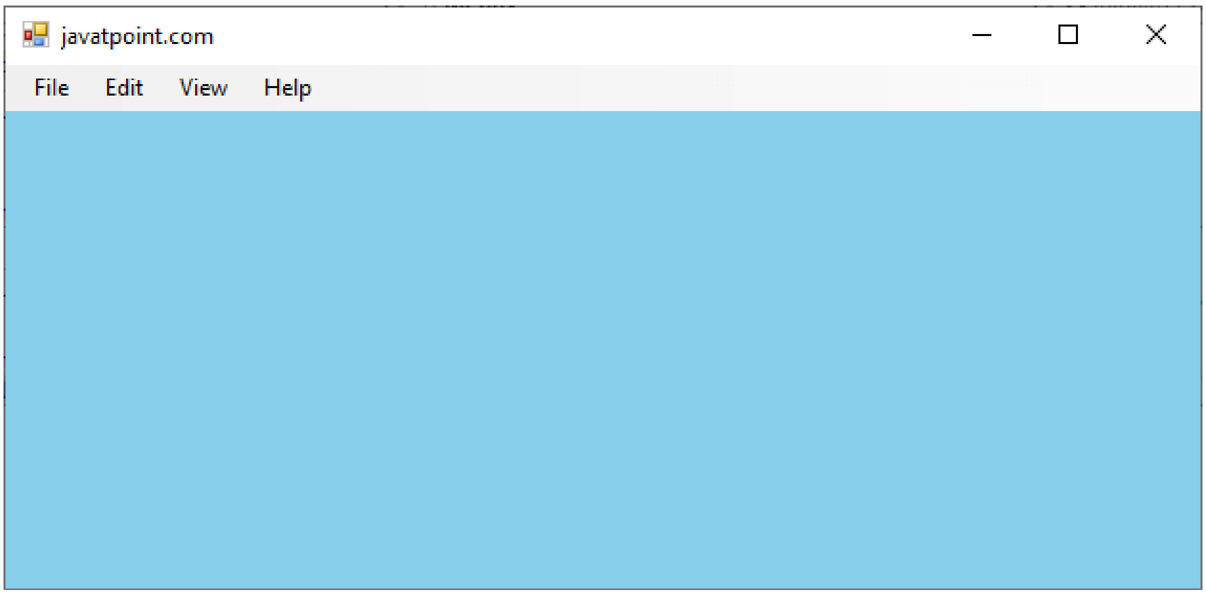
Properties
FileToolStripMenuItem System.W

Text	File
TextAlign	MiddleCenter
TextDirection	Horizontal
TextImageRela	ImageBeforeTex
Behavior	
AutoSize	True
AutoToolTip	False
CheckOnClick	False
DoubleClickEn	False
Enabled	True
ToolTipText	
Visible	True
Data	
(ApplicationSe	
DropDown	(none)
DropDownIten	(Collection)

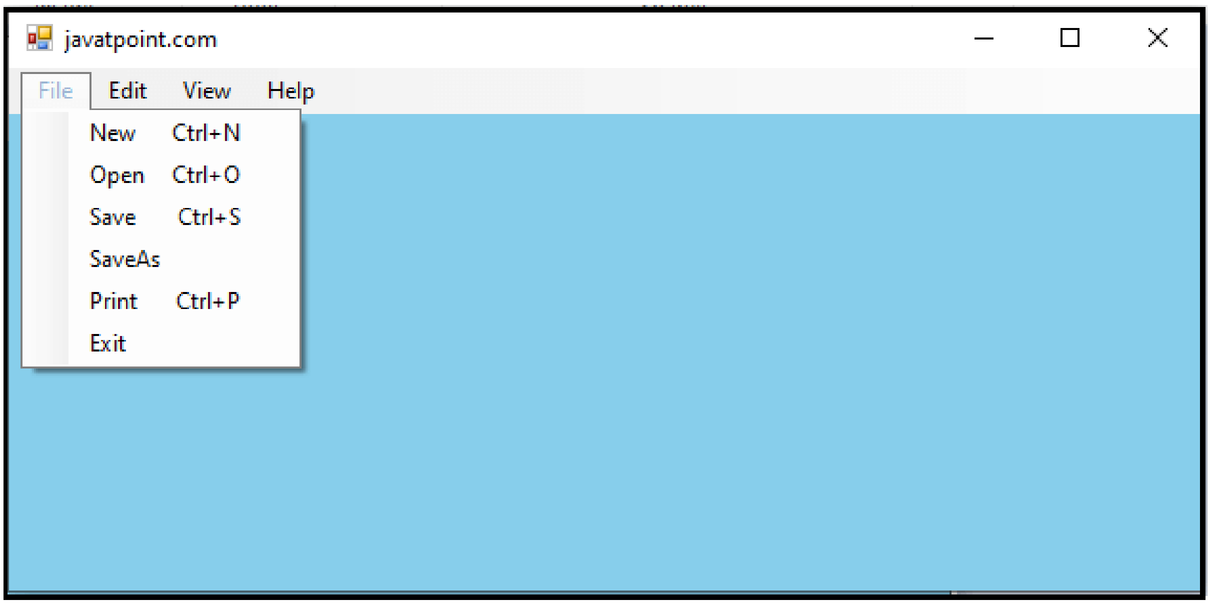
Menus.vb

```
Public Class Menus
  Private Sub Menus_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Me.Text = "javatpoint.com" 'set the title of the bar
    BackColor = Color.SkyBlue
  End Sub


  Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles ExitT
oolStripMenuItem.Click
      Me.Dispose() ' exit from the form
  End Sub
End Class
```

Output:



Click on the File menu that shows the multiple options related to files.


javatpoint.com
File
Edit
View
Help
New Ctrl+N
Open Ctrl+O
Save Ctrl+S
SaveAs
Print Ctrl+P
Exit