

Software Design & Architecture

Lab Manual

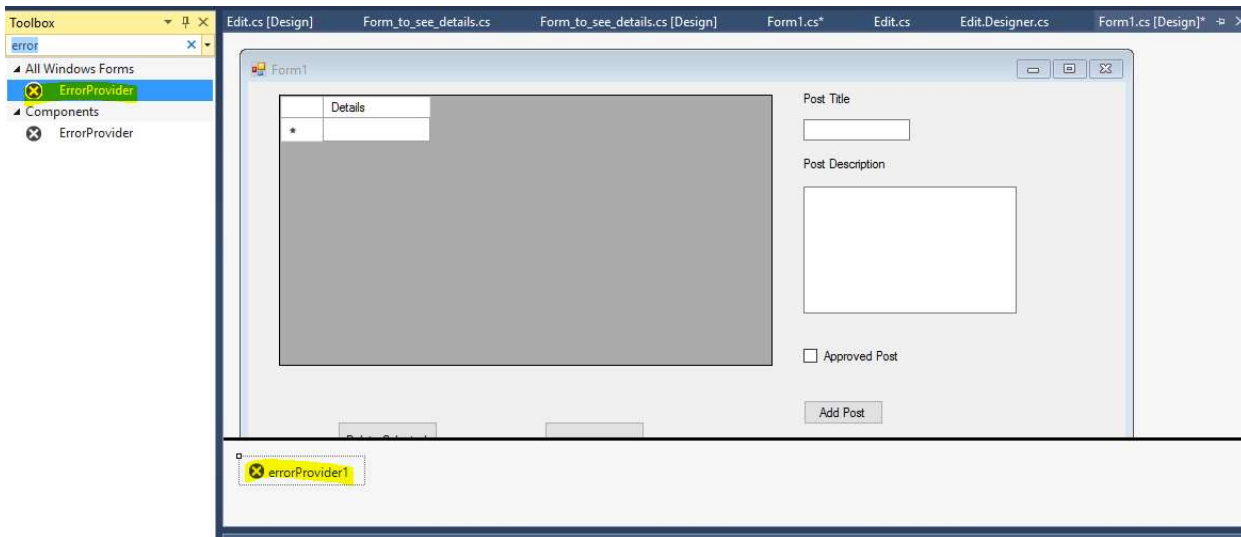
Lab 4

Form validation:

Post title and Description are mandatory to store object in list, without filling this cannot proceed further.

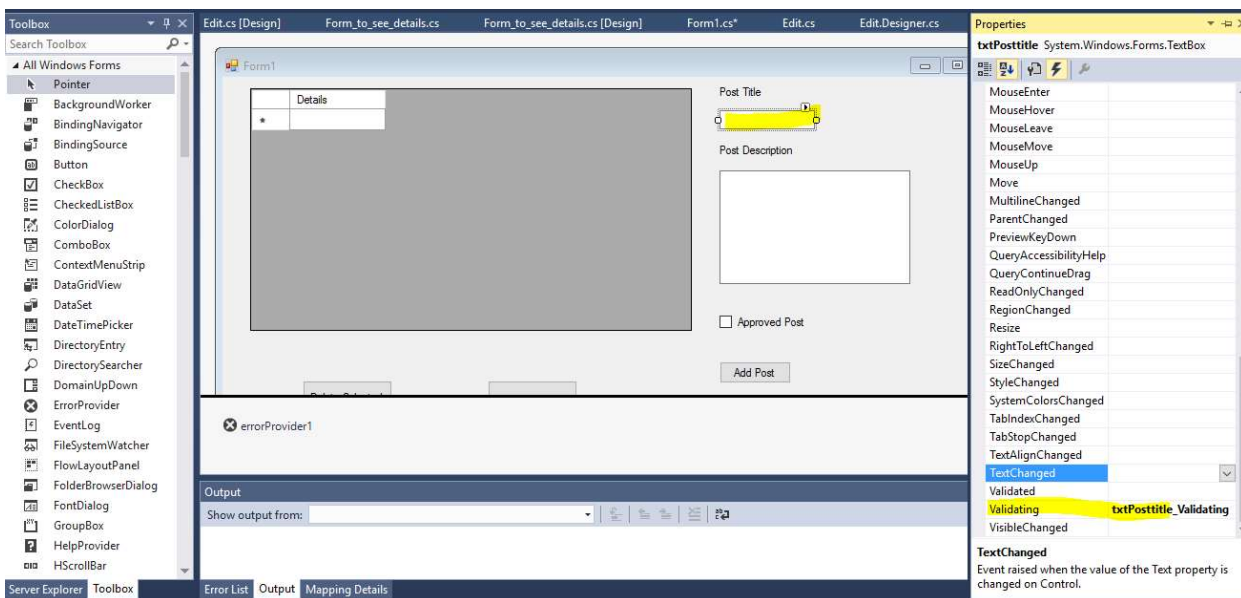
Step :1

Get “error provider” from toolbox



Step 2:

Select textbox ->go to property->events>double click on validating



Validating function:

```

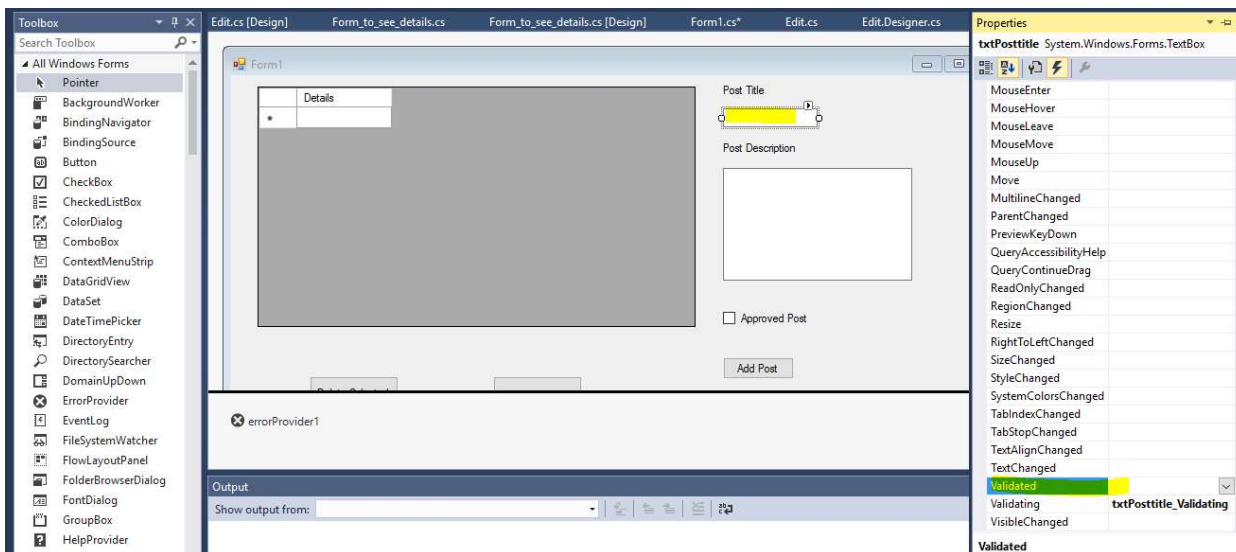
    }
}

1 reference
private void txtPosttitle_Validating(object sender, CancelEventArgs e)
{
    if (string.IsNullOrEmpty(txtPosttitle.Text)){
        e.Cancel = true;
        errorProvider1.SetError(txtPosttitle,"enter post title");
    }
    else
    {
        e.Cancel = false;
        errorProvider1.SetError(txtPosttitle,null);
    }
}
}

```

If post title textbox is empty it will cancel event and errorprovider will show message **else** when post title is mentioned it will show nothing in error provider and cancel event will remain false.

Step 3:



Click on textbox go to properties ->select validated event

```

}

1reference
private void txtPosttitle_Validating(object sender, CancelEventArgs e)
{
    if (string.IsNullOrEmpty(txtPosttitle.Text)){
        e.Cancel = true;

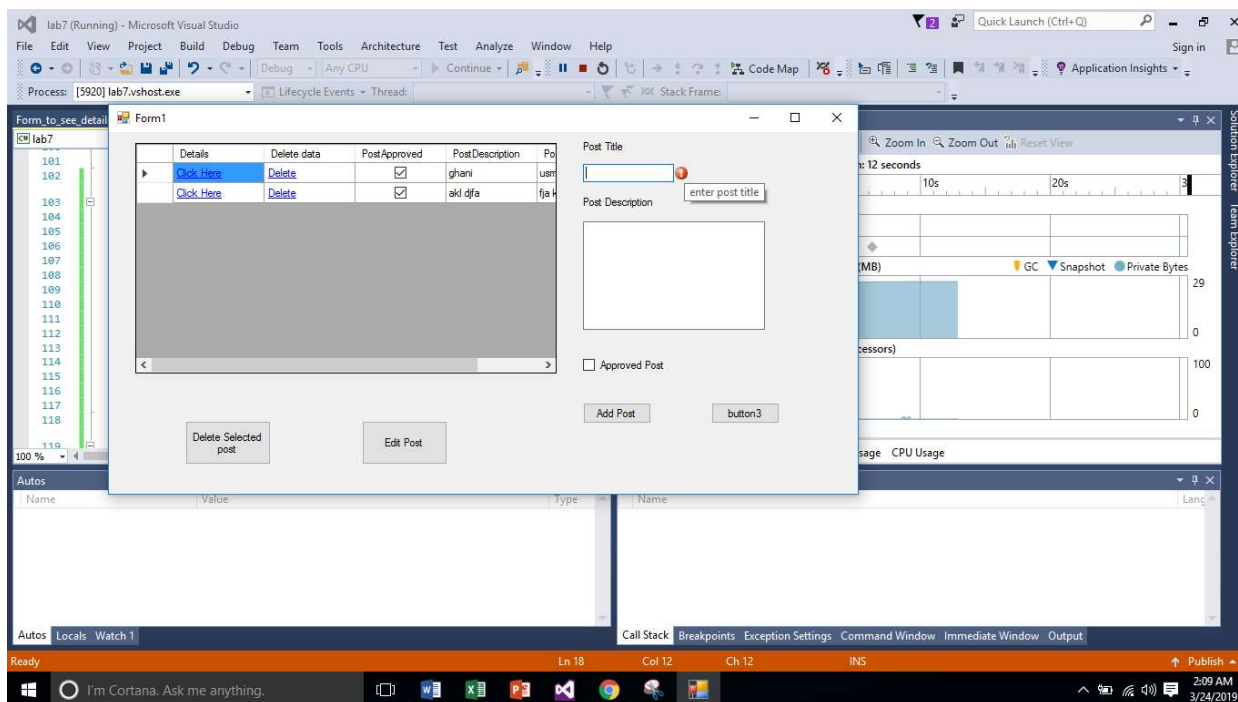
        errorProvider1.SetError(txtPosttitle,"enter post title");
    }
    else
    {
        e.Cancel = false;
        errorProvider1.SetError(txtPosttitle,null);
    }
}

//validated function will call automatically when validating functino set e.cancel set as true
1reference
private void txtPosttitle_Validated(object sender, EventArgs e)
{
    errorProvider1.SetError(txtPosttitle,null);
}

1reference
private void button3_Click(object sender, EventArgs e)

```

OUTPUT:



Make validation on description as well.

Note:

Each time when you make new function on server you have to update reference at client side.

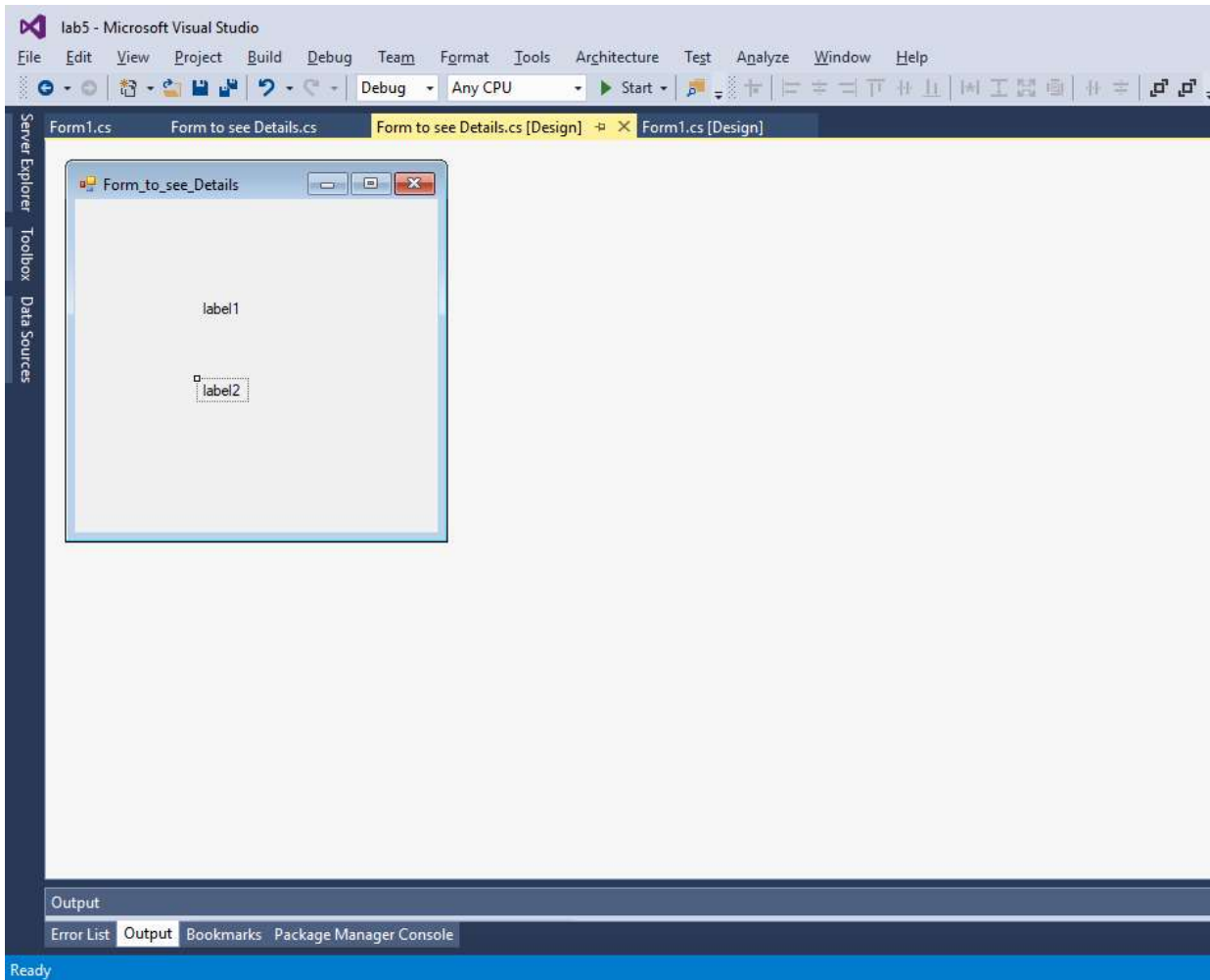
You just need to right click on server name and thne click on update server.

Objective: When User Click on “Click Here” link in first Colum of DataGridView new form will open and show post title and post description.

Solution:

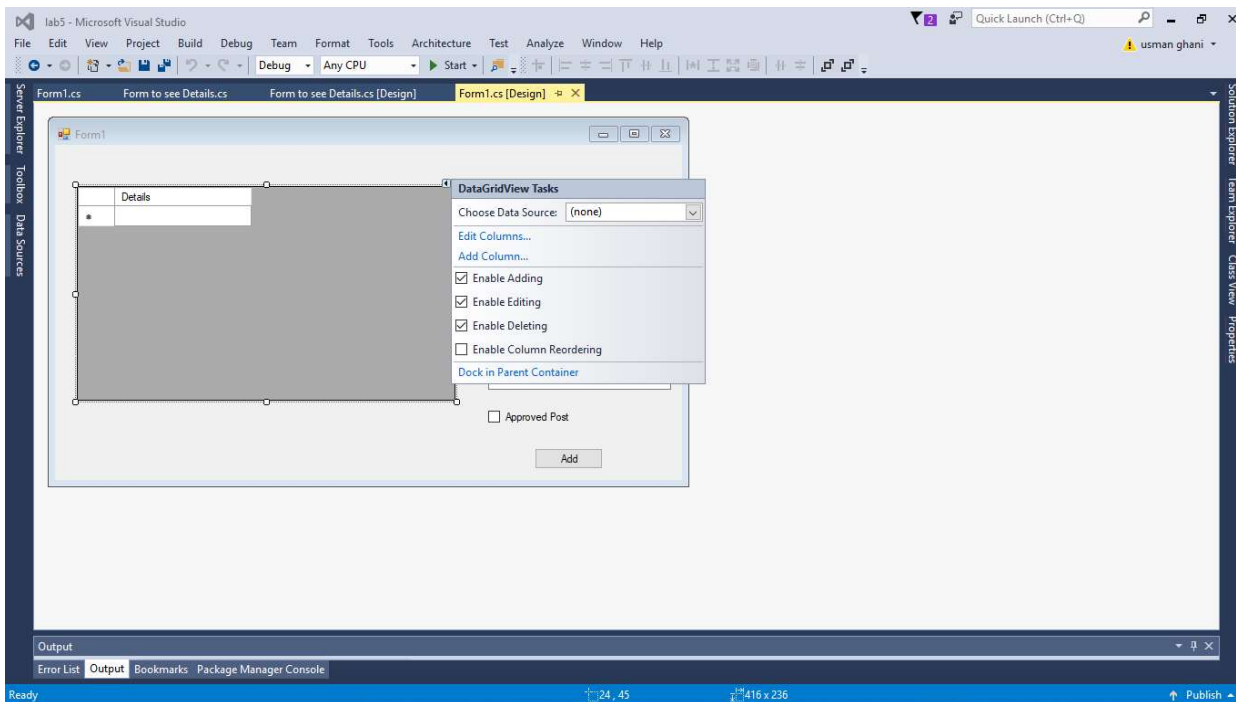
Extend lab 4 and add 2nd form

Form will contain two labels “label1” and “label2”

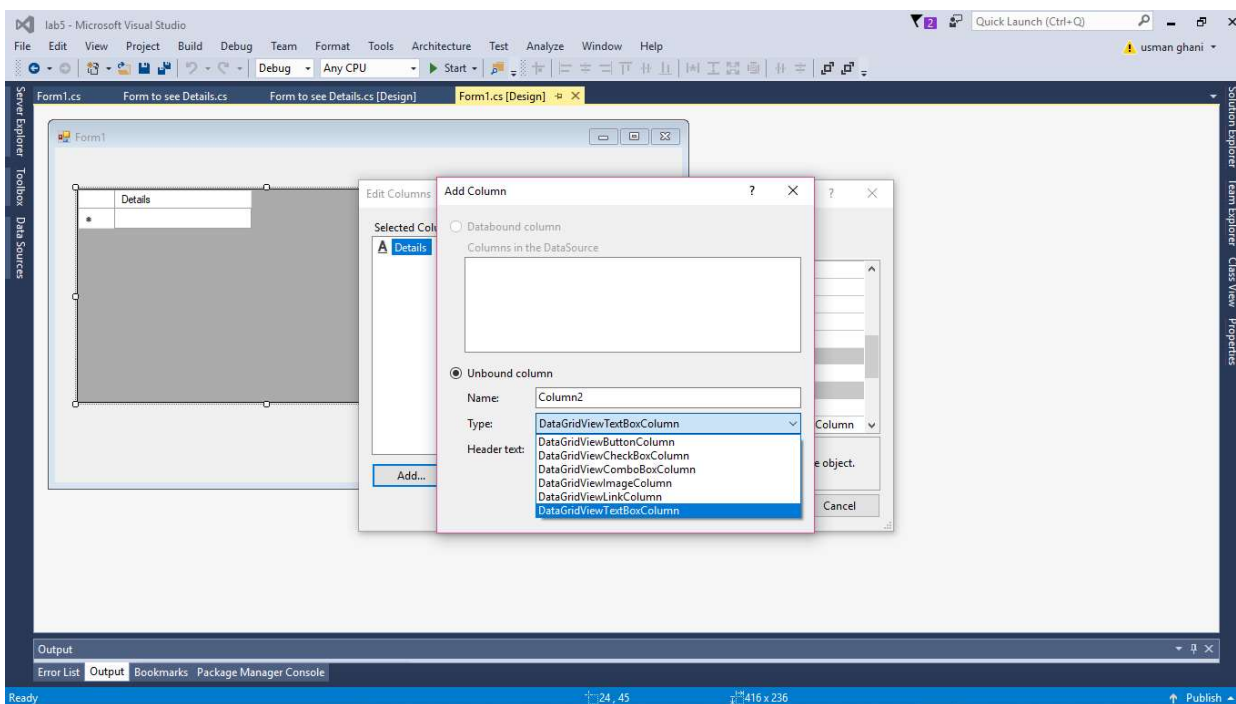


Step 2:

Go to Form1



Click on Gridview Task and select edit column

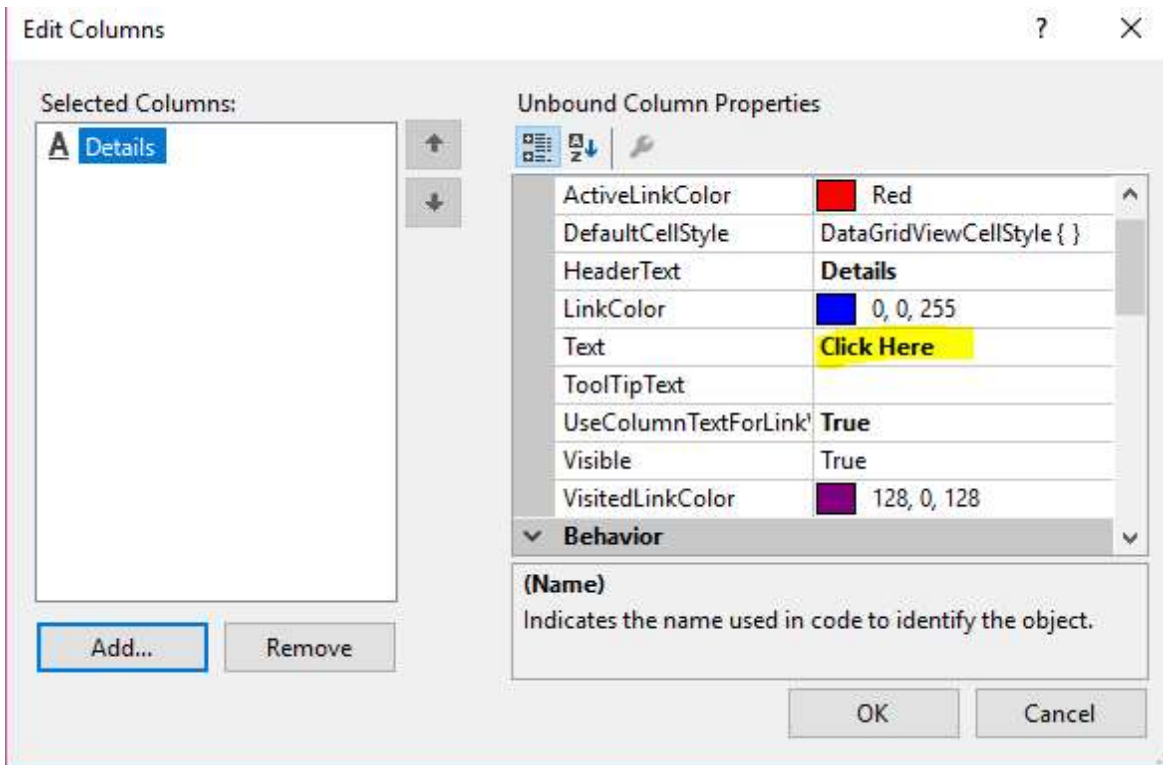


Select Column name and type

Type: select DataGridviewLinkColumn

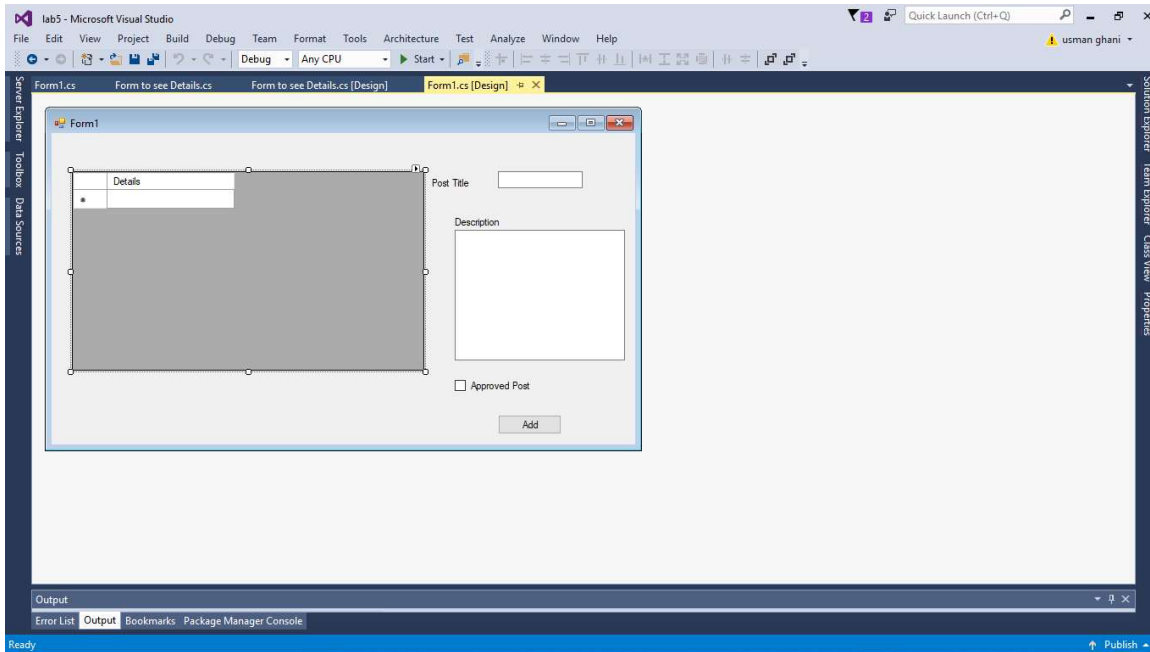
Step 3

Add Text Click Here



Step 4:

Double click on Datagridview



```

    }

    1 reference
    private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
    {
        ServiceReference1.Service1Client myserver = new ServiceReference1.Service1Client();

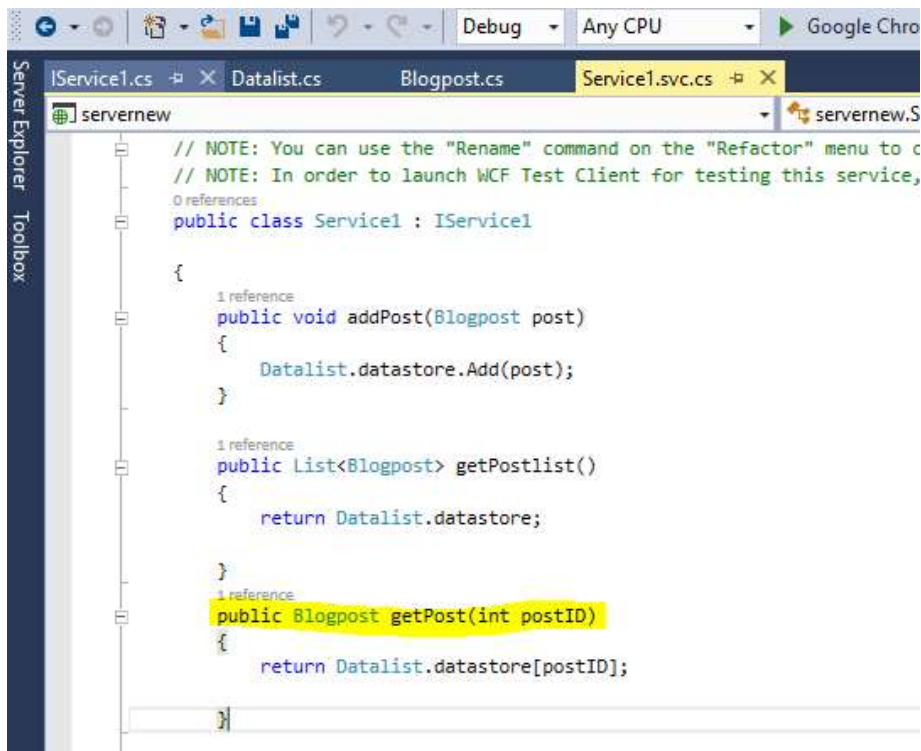
        if (e.ColumnIndex == 0)
        {
            ServiceReference1.Blogpost post = myserver.getPost(e.RowIndex);
            Form_to_see_Details abc = new Form_to_see_Details(post);
            abc.Show();
        }
    }
}
}
}

```

Open server and make function and in service1.svc.cs and add prototype in IService1.cs
getpost(e.RowIndex) is defined in Service1.svc.cs.

Step 5:

Functions Definitions



```

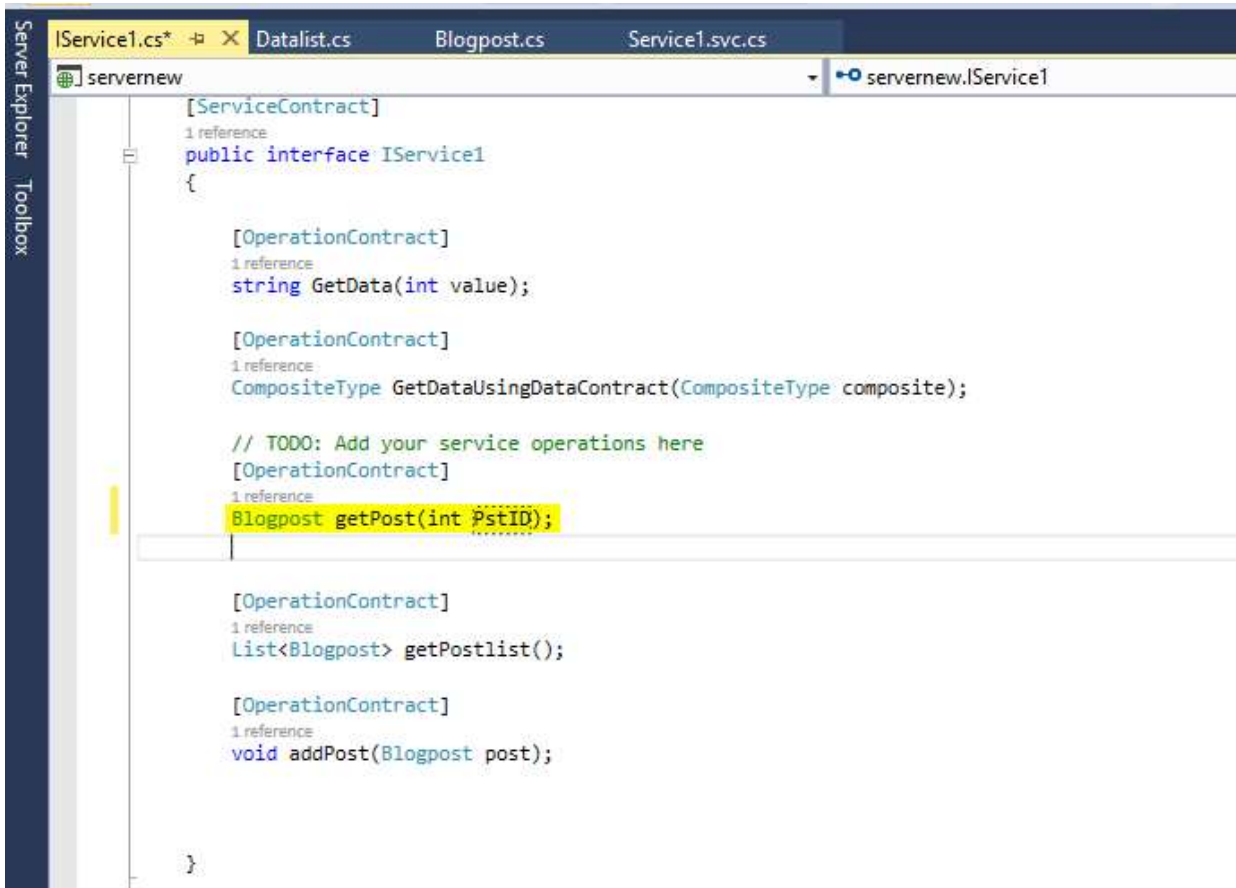
// NOTE: You can use the "Rename" command on the "Refactor" menu to c
// NOTE: In order to launch WCF Test Client for testing this service,
0 references
public class Service1 : IService1
{
    1 reference
    public void addPost(Blogpost post)
    {
        Datalist.datastore.Add(post);
    }

    1 reference
    public List<Blogpost> getPostlist()
    {
        return Datalist.datastore;
    }

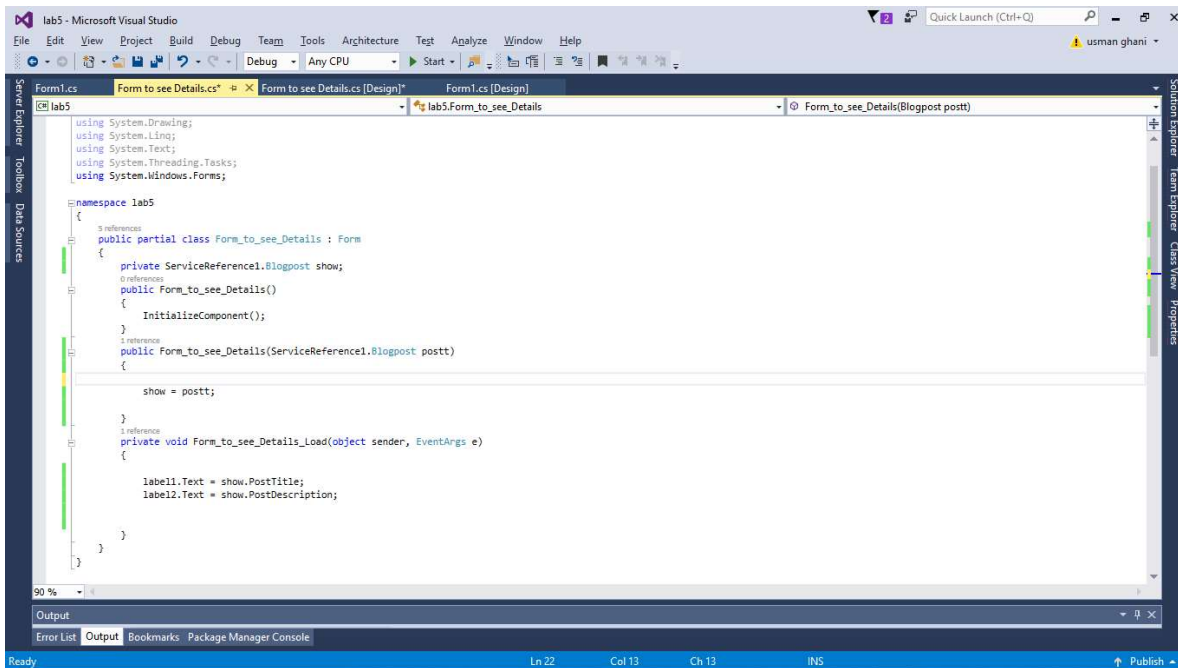
    1 reference
    public Blogpost getPost(int postID)
    {
        return Datalist.datastore[postID];
    }
}

```

Function Prototypes:



Form 2



Constructor:


```

namespace lab5
{
    5 references
    public partial class Form_to_see_Details : Form
    {
        0 references
        private ServiceReference1.Blogpost show;
        0 references
        public Form_to_see_Details()
        {
            InitializeComponent();
        }
        1 reference
        public Form_to_see_Details(ServiceReference1.Blogpost postt)
        {
            show = postt;
        }
        1 reference
        private void Form_to_see_Details_Load(object sender, EventArgs e)
        {
            label11.Text = show.PostTitle;
            label12.Text = show.PostDescription;
        }
    }
}

```

When form load event will occur it will get posttitle and postdescription and show in labels.

Final Step:

```

lab5 lab5.Form_to_see_Details

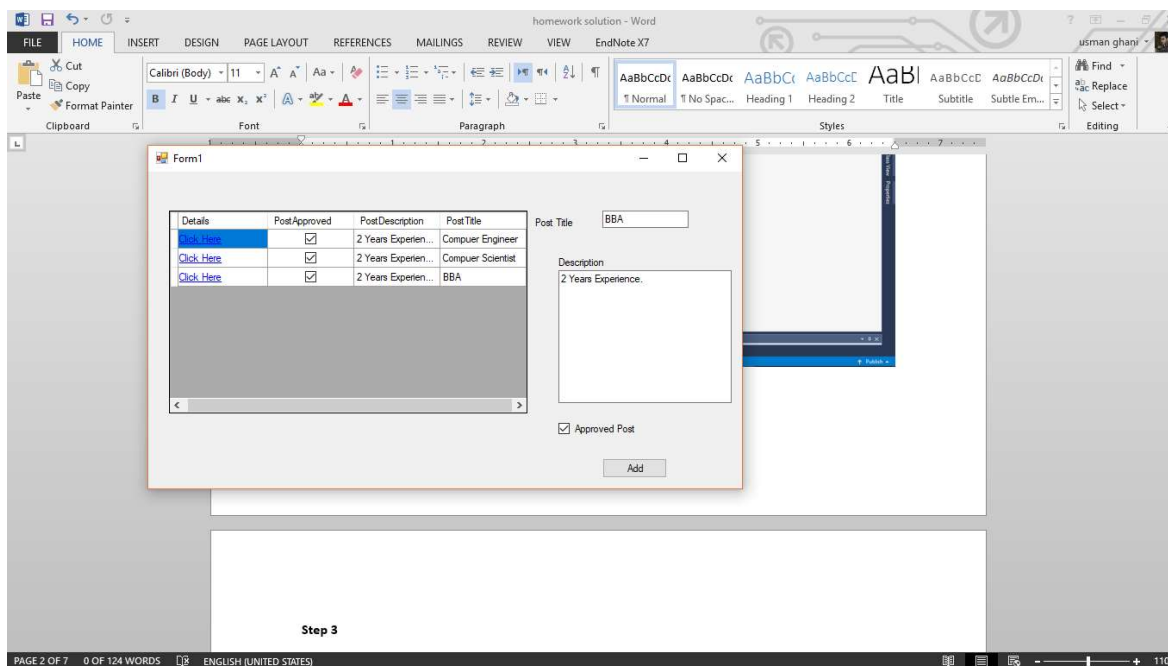
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

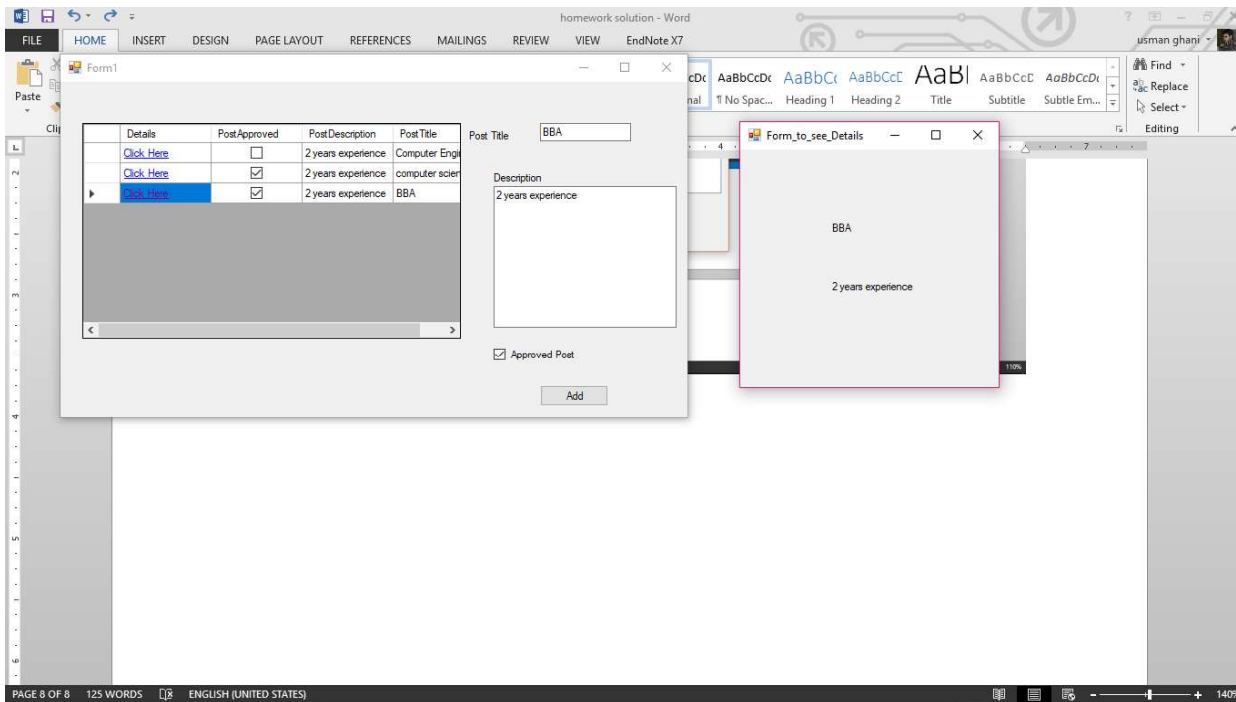
namespace lab5
{
    3 references
    public partial class Form_to_see_Details : Form
    {
        private ServiceReference1.Blogpost show;
        0 references
        public Form_to_see_Details()
        {
            InitializeComponent();
        }
        1 reference
        public Form_to_see_Details(ServiceReference1.Blogpost postt)
        {
            InitializeComponent();
            show = postt;
        }
        1 reference
        private void Form_to_see_Details_Load(object sender, EventArgs e)
        {
            label1.Text = show.PostTitle;
            label2.Text = show.PostDescription;
        }
    }
}

```

Add initializeComponent() in constructor to call the functionalities of form.

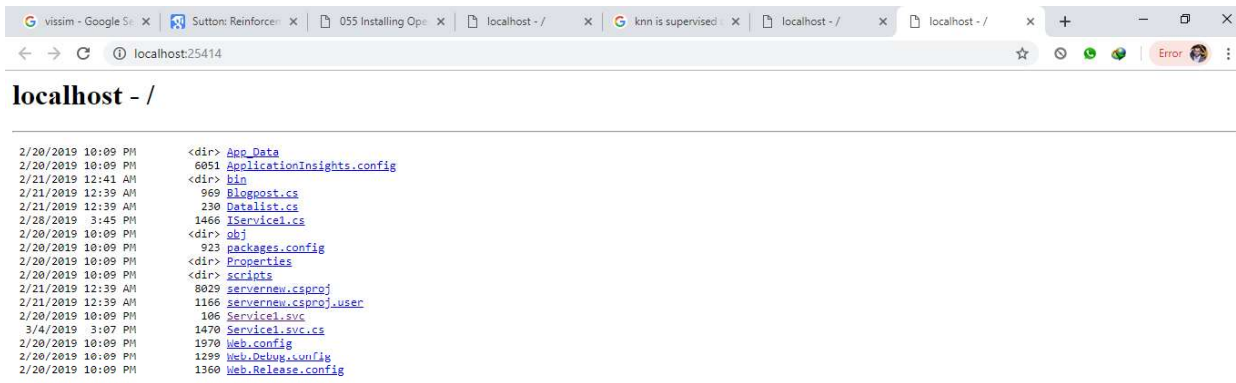
Output:





Additional information:

How to get Reference of server:



Click on service1.svc

Service1 Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following syntax:

```
svcutil.exe http://localhost:25414/Service1.svc?wsdl
```

You can also access the service description as a single file:

```
http://localhost:25414/Service1.svc?singleWsdl
```

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to call the Service. For example:

C#

```
class Test
{
    static void Main()
    {
        Service1Client client = new Service1Client();

        // Use the 'client' variable to call operations on the service.

        // Always close the client.
        client.Close();
    }
}
```

Visual Basic

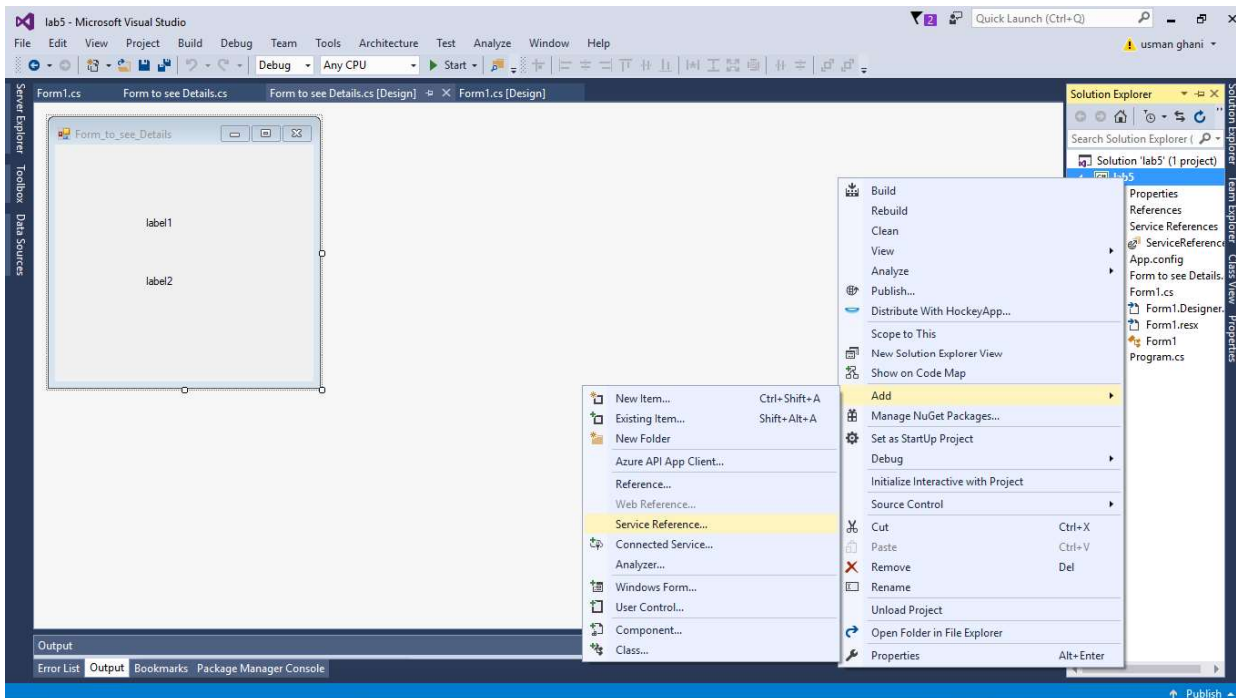
```
Class Test
    Shared Sub Main()
        Dim client As Service1Client = New Service1Client()
        ' Use the 'client' variable to call operations on the service.

        ' Always close the client.
        client.Close()
    End Sub
End Class
```

Copy link: localhost:25414/Service1.svc

2.

Add link to client side



Rightclick on project>>add>>service reference

Add Service Reference?×

To see a list of available services on a specific server, enter a service URL and click Go. To browse for available services, click Discover.

Address:

http://localhost:25414/Service1.svc

Go

Discover

Services:

Operations:

Namespace:

ServiceReference2

Advanced...

OK

Cancel

paste link and write name.

77 | Page