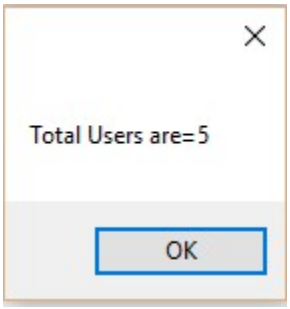
Lab 2

Implementation of class using C# Web Forms

Register users by saving his/ her information and show total registered users using a button



Code of Form:

```
public partial class Form1 : Form
  {
    public Form1()
    {
      InitializeComponent();
    }

    private void cmdLogin_Click(object sender, EventArgs e)
    {
      StudentDTO dto = new StudentDTO();
      dto.Name = txtUserName.Text;
      dto.Password = txtPassword.Text;
      dto.Age = txtAge.Text;
      StudentBL.register(dto);
    }
```

```
    private void button1_Click(object sender, EventArgs e)
    {
        int total= StudentDL.getTotalRegisterStudents();
        MessageBox.Show("Total Users are=" + total);
    }
}
```

Other three classes are as follows:

1. **DTO:**

```
class StudentDTO
  {
    private String name1;
    private String age1;
    private String password1;

    public string Name
    {
        get
        {
            return name1;
        }

        set
        {
            name1 = value;
        }
    }

    public string Age
    {
        get
        {
            return age1;
        }

        set
        {
            age1 = value;
        }
    }

    public string Password
    {
        get
        {
            return password1;
        }

        set
        {
            password1 = value;
        }
    }
```

```
}
```

2. **DL**

```
class StudentDL
  {
    private static List<StudentDTO> students = new List<StudentDTO>();
    public static void addUsers(StudentDTO sdto)
    {
      students.Add(sdto);
    }

    public static StudentDTO getUsersByName(String name) {

      return students.Find(sd => sd.Name == name);
    }

    public static int getTotalRegisterStudents() {

      return students.ToList().Count;
    }
  }
```

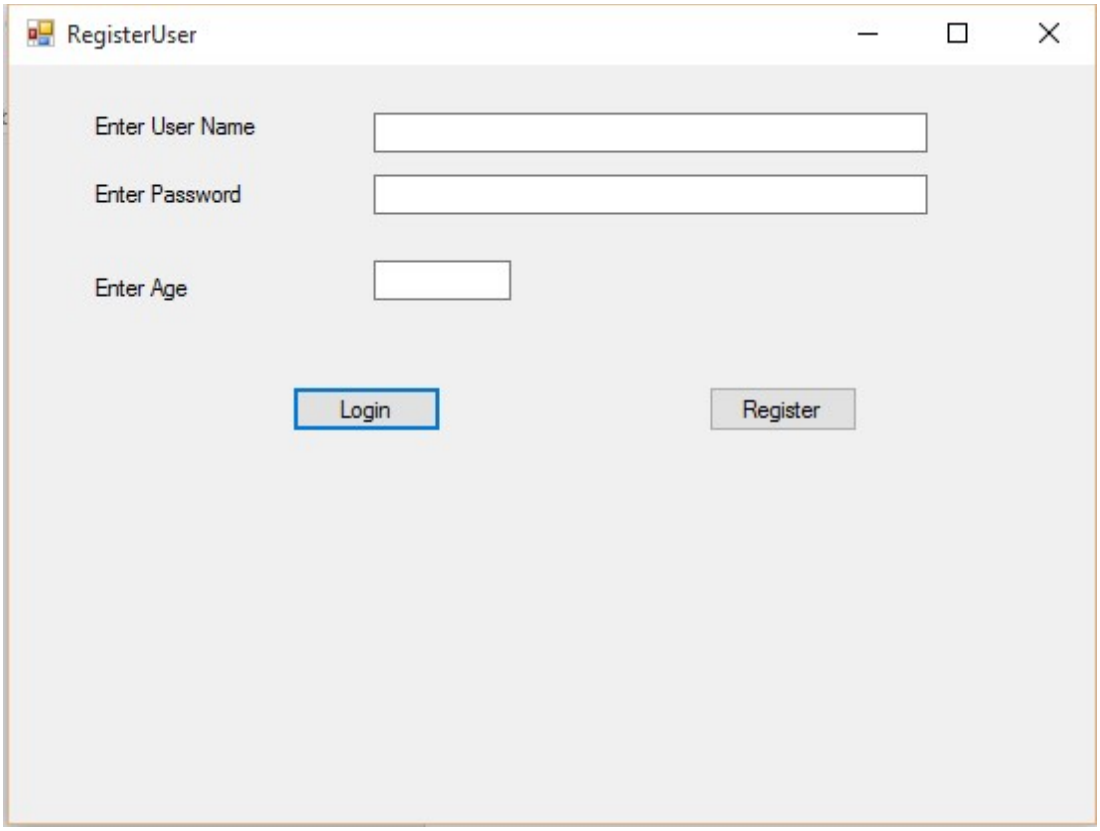
3. **BL:**

```
class StudentBL
  {
    StudentDTO dto2;
    StudentBL(StudentDTO dto2) {
      this.dto2 = dto2;
    }


    public static StudentBL register(StudentDTO dto2) {

      StudentDL.addUsers(dto2);
      StudentBL studentBL = new StudentBL(dto2);
      return studentBL;
    }
  }
```

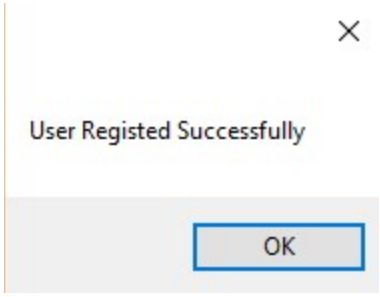
Code the scenario given below in C#.

Register users using user name and password and age, after successful registration users will login using registered username and password, and after successful sign in age of the user will be shown which he/ she entered in the first form. (also validate form against empty as well as wrong input and show message if user successfully registered)

Form 1:



Validation and Message Printing:

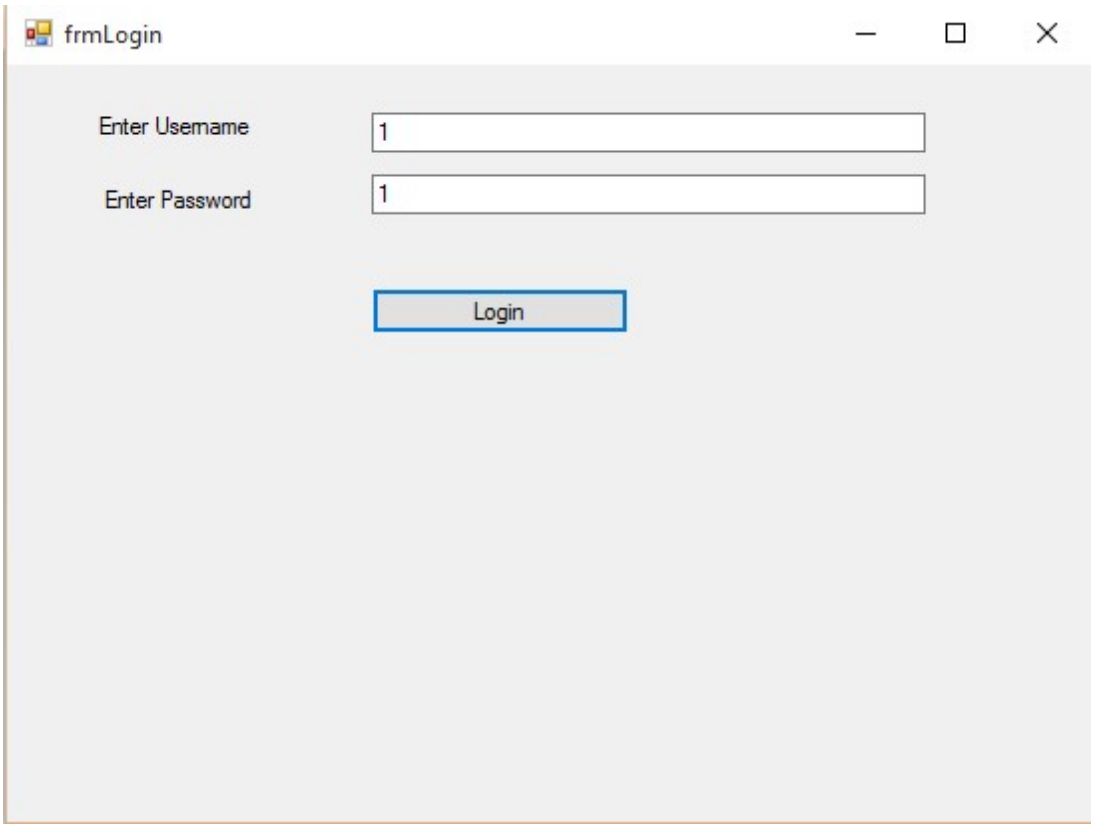


Code of Form 1:

```
private void btnRegister_Click(object sender, EventArgs e)
    {
        if (txtUserName.Text != "" && txtPassword.Text != "" && txtAge.Text != "")
        {
            StudentDTO std = new StudentDTO();
            std._name = txtUserName.Text;
            std._password = txtPassword.Text;
            std.age = Int32.Parse(txtAge.Text);
            StudentDL.addStudent(std);
            MessageBox.Show("User Registed Successfully");
            txtUserName.Text = "";
            txtPassword.Text = "";
            txtAge.Text = "";
        }
        else
            MessageBox.Show("Enter Complete Data");
```

```
    }
```

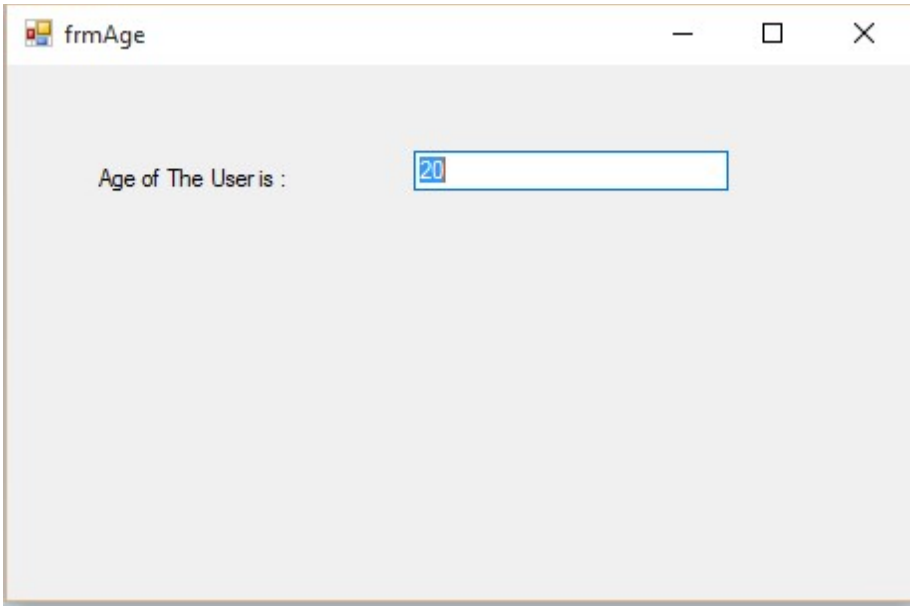
Form 2:



Code:

```
private void btnLogin_Click(object sender, EventArgs e)
    {
      if(txtUserName.Text!="" && txtPassword.Text!="")
      {
        StudentDTO std = new StudentDTO();
        std._name = txtUserName.Text;
        std._password = txtPassword.Text;
        std = StudentDL.isValidUser(std);
        if (std != null)
        {
          frmAge formAge = new frmAge(""+std._age);
          formAge.Show();

        }
        else
          MessageBox.Show("Invalid user");
      }
    }
```

Form 3:



Code:

```
public frmAge(string Age)
    {
      InitializeComponent();
      txtAge.Text = Age;
    }
```

Other two classes are as follows:

1. **DTO:**

```
class StudentDTO
  {
    public string _name;
    public string _password;
    public int _age;
    // setter getter for Name
    public string name
    {
      set
      {
        _name = value;

      }
      get
      {
        return _name;
      }
    }
    // setter getter for password
    public string password
    {
      set
      {
        _password = value;
```

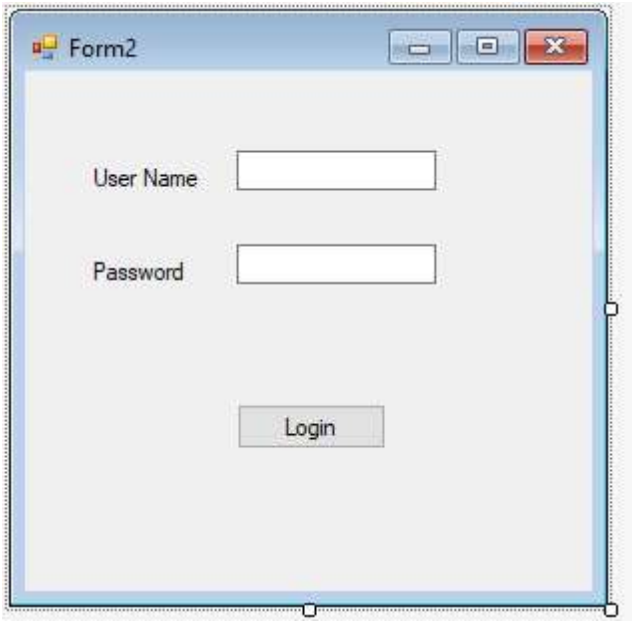
```
        }
        get
        {
            return _password;
        }
    }
    // setter getter for Age
    public int age
    {
        set
        {
            _age = value;
        }
        get
        {
            return _age;
        }
    }

}
```

2. DL:

```
class StudentDL
  {
    public static ArrayList users = new ArrayList();

    public static void addStudent(StudentDTO std)
    {
        users.Add(std);
    }
    public static StudentDTO isValidUser(StudentDTO std)
    {
        foreach (StudentDTO u in users)
            if (u._name.Equals(std._name) && (u._password.Equals(std._password)))
              return u;
        return null;
    }
    public static int getTotal()
    {
        return users.Count;
    }
  }
```

Program 1:

- Change the Login Page such that it can check 10 different users and password and allow them to login.



Event behind button click(Login):

```
ooadLab2 | ooadLab2.Form1 | button1_Click_1(object sender, EventArgs e)

string[] users = new string[10] { "usman", "farhan", "subhan", "noman", "umer", "ali", "rizwan", "sajad", "shahbaz", "arbaz" };
string[] password = new string [10]{ "123", "456", "789", "123", "456", "789", "123", "456", "789", "123" };

for (int i = 0; i < 10; i++)
{
    if ((textBox1.Text == users[i]) && (textBox2.Text == password[i]))
    {
        label3.BackColor = System.Drawing.Color.Green;
        label3.Text = "Validate user";
        break;
    }



    else

    {

        label3.BackColor = System.Drawing.Color.Red;
        label3.Text = "inValidate user";
    }
}
```

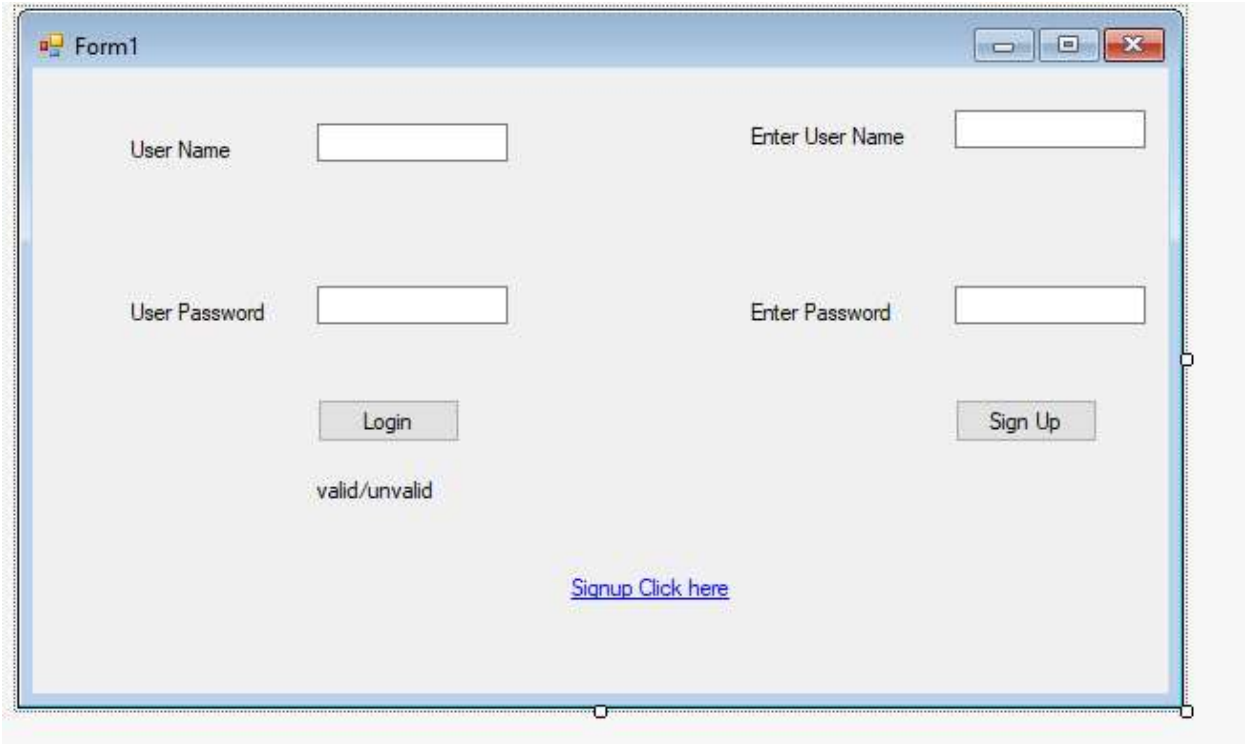
100 %

Output

Show output from: IntelliSense

Program 2:

- Extend the login page and add signup functionality into it so that a new user can be registered and our previous program allow that user to login.



Step 1:

Make a Global variable username and password

```
public partial class Form1 : Form
{
    2 references
    public Form1()
    {
        InitializeComponent();
    }

    string[] username = new string[10];
    string[] password = new string[10];
    int u = 0;

    0 references
    private void button1_Click(object sender, EventArgs e)
    {

    }
```

Step 2:

Code behind signup event

```
1 reference
private void button1_Click_2(object sender, EventArgs e)
{

    username[u] = textBox3.Text;
    password[u] = textBox4.Text;
    u = u + 1;


}

1 reference
```

Step 3:

Code behind Login event:

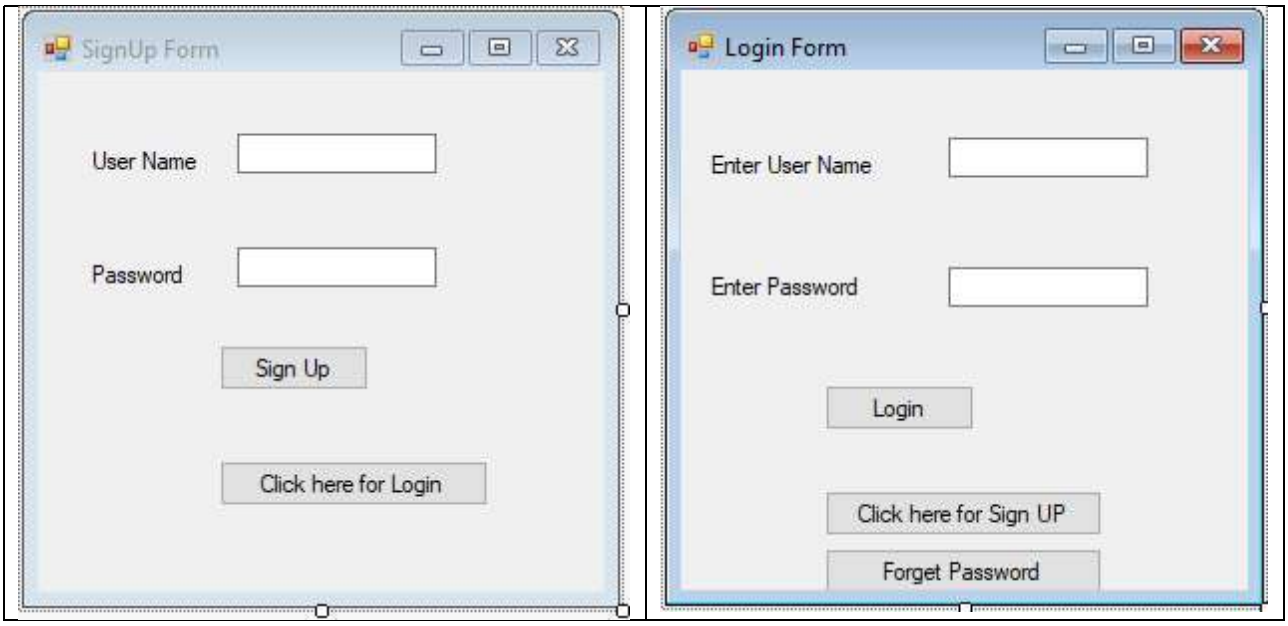
```
for (int i = 0; i < 10; i++)
{
    if ((textBox1.Text == username[i]) && (textBox2.Text == password[i]))
    {
        label3.BackColor = System.Drawing.Color.Green;
        label3.Text = "Validate user";
        break;
    }

    else

    {

        label3.BackColor = System.Drawing.Color.Red;
        label3.Text = "inValidate user";
    }

```

Program 3:

- Separate Login and Signup Pages. Make the Navigation Flow between them.

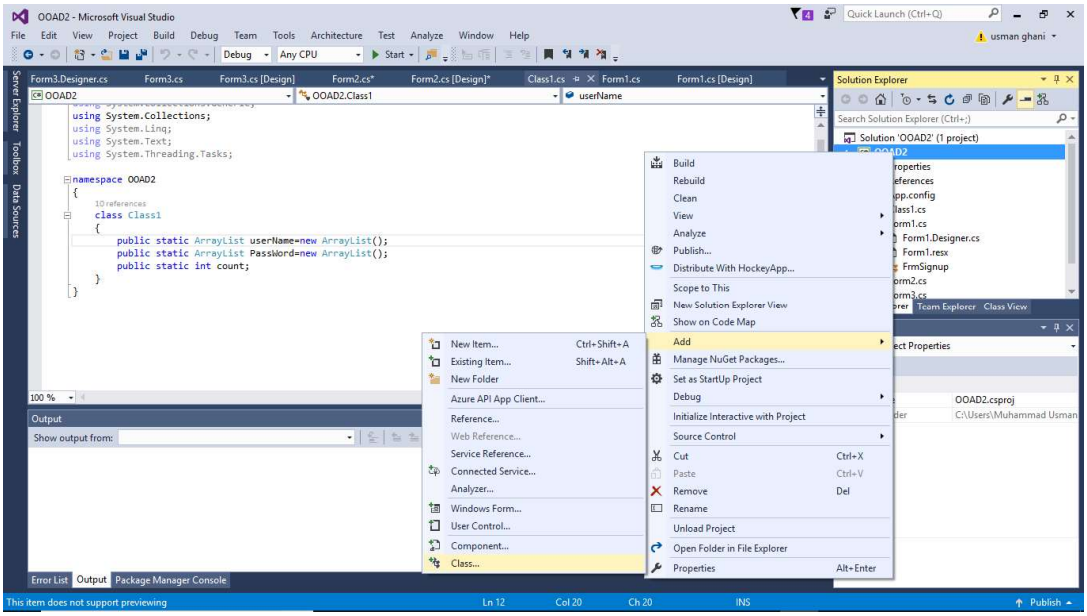
Step 1:

Design Forms:



Step 2 :

Add “Class.cs” by selecting project



Step 3:

Add “using system.collection” to Make ArrayList of username and password.

Make these ArryasLists static so that their value can be access by both forms.

Static variables are access with the name of class name (not need to make object of class).

Step 4:

Code behind signup event:

```
1 reference
private void button1_Click(object sender, EventArgs e)

{
    Class1.userName.Add(txtUsername.Text);
    Class1.PassWord.Add(txtPassword.Text);
    Class1.count = Class1.count + 1;
    MessageBox.Show("Signup successfully");

}
```

Step 5:

Code behind “click here for login” event on signup form:

```
1 reference
private void button2_Click(object sender, EventArgs e)
{
    Form2 login = new Form2();
    login.Show();
}
```

Step 6:

Code behind “login” event:

```
1 reference
private void button1_Click(object sender, EventArgs e)
{

    for (int i=0;i<Class1.count; i++)
    {

        if (txtUsername.Text == (string)Class1.userName[i] && textPassword.Text == (string)Class1.PassWord[i])
        {

            lblstatus.Text = "Valid user";

            break;
        }

        else
            lblstatus.Text = "Invalid user";
    }
}
```

Step 7:

Code behind “signup” event on Login Form(for navigation):

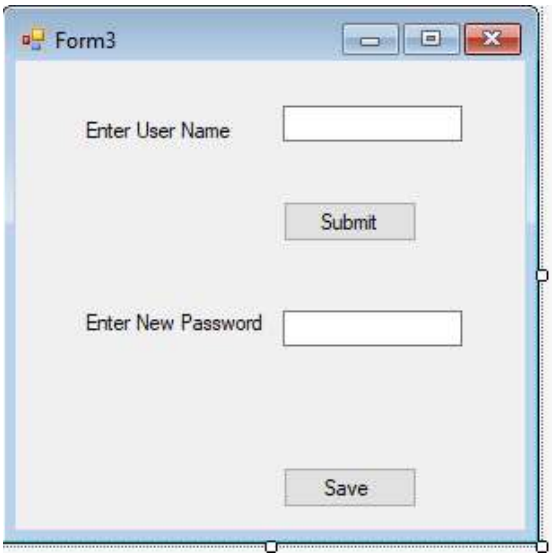
```
1 reference
private void button2_Click(object sender, EventArgs e)
{
    FrmSignup signup = new FrmSignup();
    signup.Show();
}
```

Program 4:

- If you successfully login, take it to another another page call it home page where he can view his username and change his password. Once user changed his password, he can login with new password

Step 1:

Extend the program3; add 3rd form (as a home page)



Step 2:

Verify the valid user by getting his/her user name (code behind submit button)

```
1 reference
private void btnSubmit_Click(object sender, EventArgs e)
{
    for (int k = 0; k < Class1.count; k++)
    {

        if (txtUsername.Text == (string)Class1.userName[k])
        {

            lblvalidation.Text = "Enter Your newpassword";
            str_index = k;

            break;
        }

        else
            lblvalidation.Text = "Invalid user Name";
    }
```

Get index value in str_index(global variable) so that we can change password according to user name.

Step 3:

Code behind “save” event

```
{

    int p;
    p = str_index;
    Class1.PassWord[p] = txtPassword.Text;
}
```

After clicking on save password you can update new password.