PRACTICAL NO.07

Write the program for the following:

a. Create a web application to demonstrate the use of different types of Cookies.
Code:

WebForm1.aspx :

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="Cookies.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body style="height: 192px">
   <form id="form1" runat="server">
      <div style="height: 175px">
         <asp:Label ID="Label1" runat="server" Text="Cookie Example :"></asp:Label>
         <br />
         <br />
         <asp:Label ID="Label2" runat="server" Text="Eneter Name:"></asp:Label>
    
         <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
         <br />
         <br />
         <asp:Label ID="Label3" runat="server" Text="Enter Password:"></asp:Label>
         <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
         <br />
         <br />

         <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Next" />
      </div>
   </form>
</body>
</html>
```

WebForm1.aspx.cs :

```
using System;
using System.Web;

namespace Cookies
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
      }
      protected void Button1_Click(object sender, EventArgs e)
      {
         HttpCookie userInfo = new HttpCookie("userInfo");
         userInfo["UserName"] = TextBox1.Text;
         userInfo["Password"] = TextBox2.Text;
         userInfo.Expires = DateTime.Now.AddSeconds(5);
         Response.Cookies.Add(userInfo);
         Response.Redirect("WebForm2.aspx");
      }
   }
}
```

WebForm2.aspx :

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs"
Inherits="Cookies.WebForm2" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body style="height: 229px">
    <form id="form1" runat="server">
        <div style="height: 235px">
            <asp:Label ID="Label1" runat="server" Text="Name is "></asp:Label>
            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
            <br />
            <br />
            <asp:Label ID="Label2" runat="server" Text="Password is"></asp:Label>
            <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
        </div>
    </form>
</body>
</html>
```

WebForm2.aspx.cs :
```csharp
using System;
using System.Web;

namespace Cookies
{
    public partial class WebForm2 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            HttpCookie reqCookies = Request.Cookies["userInfo"];
            TextBox1.Text = reqCookies["UserName"].ToString();
            TextBox2.Text = reqCookies["Password"].ToString();
        }
    }
}
```

Output:

Cookie Example :

Eneter Name: Ram

Enter Password: admin@123

Next

Name is Ram

Password is admin@123

b. Create a web application to demonstrate Form Security and Windows Security with proper

Authentication and Authorization properties.
Code:
Web.config :

```xml
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  https://go.microsoft.com/fwlink/?LinkId=169433
  -->
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.7.2" />
    <httpRuntime targetFramework="4.7.2" />

        <!-- Configure Forms Authentication -->
        <authentication mode="Forms">
            <forms loginUrl="Login.aspx" defaultUrl="Default.aspx">
                <credentials passwordFormat="Clear">
                    <!-- Users with their respective passwords -->
                    <user name="Admin" password="Admin"/>
                    <user name="User" password="123456"/>
                </credentials>
            </forms>
        </authentication>

        <!-- Authorization rules -->
        <authorization>
            <!-- Allow specific users to access secured pages -->
            <allow users="Admin,User"/>
            <!-- Deny anonymous users (not logged in) -->
            <deny users="?" />
        </authorization>
  </system.web>
  <system.codedom>
    <compilers>
      <compiler language="c#;cs;csharp" extension=".cs"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:default /nowarn:1659;1699;1701" />
      <compiler language="vb;vbs;visualbasic;vbscript" extension=".vb"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:default /nowarn:41008 /define:_MYTYPE=\&quot;Web\&quot;
/optionInfer+" />
    </compilers>
  </system.codedom>
</configuration>
```

Login.aspx :

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Login.aspx.cs"
Inherits="form_security.Login" %>

<!DOCTYPE html>
<html>
<head>
  <title>Login</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>Login Page</h2>
      Username: <asp:TextBox ID="TextBox1" runat="server" /><br /><br />
```

```
        Password: <asp:TextBox ID="TextBox2" runat="server" TextMode="Password" /><br
/><br />
            <asp:Button ID="Button1" runat="server" Text="Login" OnClick="Button1_Click" />
            <br /><br />
            <asp:Label ID="Label4" runat="server" ForeColor="Red" />
        </div>
    </form>
</body>
</html>
```

Login.aspx.cs:
```
using System;
using System.Web.Security;

namespace form_security
{
    public partial class Login : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
        protected void Button1_Click(object sender, EventArgs e)
        {
            string uname = TextBox1.Text;
            string pass = TextBox2.Text;

            // Use FormsAuthentication to validate the user credentials (stored in Web.config)
            if (FormsAuthentication.Authenticate(uname, pass))
            {
                // If credentials are valid, redirect the user to the default page (WebForm13.aspx)
                FormsAuthentication.RedirectFromLoginPage(uname, false);
            }
            else
            {
                // Show error message if credentials are invalid
                Label4.Text = "Login Details are Invalid";
            }
        }
    }
}
```

Default.aspx:
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="form_security.Default" %>

<!DOCTYPE html>
<html>
<head>
    <title>Secured Page</title>
</head>
<body>
    <h1>Welcome, <%= Context.User.Identity.Name %>!</h1>
    <a href="Logout.aspx">Logout</a>
</body>
</html>
```

Default.aspx.cs:
```
using System;
using System.Web.Security;

namespace form_security
{
    public partial class Default : System.Web.UI.Page
    {
```

```csharp
    protected void Page_Load(object sender, EventArgs e)
    {
        // Ensure the page is only accessed by authenticated users
        if (!User.Identity.IsAuthenticated)
        {
            Response.Redirect("WebForm12.aspx");
        }
    }
  }
}
```

Logout.aspx:
```
<%@ Page Language = "C#" AutoEventWireup="true" CodeBehind="Logout.aspx.cs"
Inherits="form_security.Logout" %>
```

Logout.aspx.cs:

```csharp
using System;
using System.Web.Security;

namespace form_security
{
    public partial class Logout : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            // Log out the user and redirect them to the login page
            FormsAuthentication.SignOut();
            Response.Redirect("Login.aspx");
        }
    }
}
```

Output:

Login Page — localhost:44376/Login.aspx — Username: Admin, Password: •••••, Login

Welcome, Admin! — localhost:44376/Default.aspx — Logout

Login Page — localhost:44376/Login.aspx — Username: Admin123, Password: [empty], Login — Login Details are Invalid