

## Phase-End Project

### Customer Support Logger

#### Source code:

#### HomeController:

```
using customer.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace customer.Controllers
{
    public class HomeController : Controller
    {
        customerEntities db = new customerEntities();
        public ActionResult Index()
        {
            return View();
        }
        [HttpPost]

        public ActionResult Index(UserInfo log)
        {
            var user = db.UserInfos.Where(x => x.UserId == log.UserId &&
x.passwords == log.passwords).Count();
            if (user > 0)
            {
                return RedirectToAction("Dashboard");
            }
            else
            {
                //return View();
                return RedirectToAction("Index");
            }
        }
        public ActionResult Dashboard()
```

```

    {
        custinfo c = new custinfo();
        return View(c);
    }
    [HttpPost]
    public ActionResult Dashboard(custinfo cust)
    {

        custinfo c = new custinfo();

        c.logid = Int32.Parse(Request["logid"]);
        c.customer_email = Request["customer_email"];
        c.customer_name = Request["customer_name"];
        c.log_status = Request["log_status"];
        //c.UserId = Int32.Parse(Request["UserId"]);
        c.descriptions = Request["descriptions"];
        //user.Insert(c);

        db.custinfoes.Add(c);
        db.SaveChanges();
        Response.Write("<script> alert('complaint registered successfully')</script>");

        return RedirectToAction("Index");

    }
}

```

### **Model:**

#### **UserInfo.cs:**

```

namespace customer.Models
{
    using System;
    using System.Collections.Generic;

    public partial class UserInfo
    {

```

```

        public int UserId { get; set; }
        public string Email { get; set; }
        public string passwords { get; set; }
    }
}

```

### **Custinfo.cs:**

```

namespace customer.Models
{
    using System;
    using System.Collections.Generic;

    public partial class custinfo
    {
        public int logid { get; set; }
        public string customer_email { get; set; }
        public string customer_name { get; set; }
        public string log_status { get; set; }
        public string descriptions { get; set; }
    }
}

```

### **DalTest:**

#### **customer\_support\_class.cs:**

```

using customer.Models;
using NUnit.Framework;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DalTest
{
    public class customer_support_class
    {
        [TestCase(1)]
        public void Userinfo_test(int id)
        {

```

```

        customerEntities db = new customerEntities();
        var found = db.UserInfoes.ToList();

        Assert.AreEqual(id, found[0].UserId);

        Console.WriteLine("Validation success");

    }
    [TestCase("abcd123")]
    [TestCase("suki123")]
    public void Userinfo_test1(string p)
    {
        customerEntities db = new customerEntities();
        var found = db.UserInfoes.ToList();

        Assert.AreEqual(p, found[0].passwords.ToString());
        Console.WriteLine("Validation success");

    }
    Validation v = new Validation();
    [TestCase(1, "suki123@gmail.com")]
    public void Userinfo_test2(int uid, string s)
    {

        v.check(uid, s);
    }
}

```

### **Validation.cs:**

```

using customer.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DalTest
{
    public class Validation

```

```

{
    customerEntities db = new customerEntities();
    public bool check(int uid, string s)
    {
        bool ans = false;
        var found = db.UserInfoes.ToList();
        if (found[0].UserId == uid && found[0].Email == s)
        {
            ans = true;
        }
        return ans;
    }
}

```

### Views:

#### Index.cshtml:

**@model** customer.Models.UserInfo

```

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

```

<h2>Customer Support Executive Login</h2>

**@using** (Html.BeginForm("Index", "Home", FormMethod.Post))

```

{
    @Html.AntiForgeryToken()

```

<div class="form-horizontal">

<hr />

**@Html.ValidationSummary**(true, "", new { @class = "text-danger" })

<div class="form-group">

**@Html.LabelFor**(model => model.UserId, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

```

        @Html.EditorFor(model => model.UserId, new { htmlAttributes = new
{ @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.UserId, "", new {
@class = "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Email, htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Email, new { htmlAttributes = new {
@class = "form-control" } })
        @Html.ValidationMessageFor(model => model.Email, "", new { @class
= "text-danger" })
    </div>
    </div>

<div class="form-group">
    @Html.LabelFor(model => model.passwords, htmlAttributes: new {
@class = "control-label col-md-2" })
    <div class="col-md-10">

        @Html.EditorFor(model => model.passwords, new { htmlAttributes =
new { @class = "form-control", type = "password" } })

        @Html.ValidationMessageFor(model => model.passwords, "", new {
@class = "text-danger" })
    </div>
    </div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Login" class="btn btn-default" />
    </div>
</div>
</div>
}

```

```
@section Scripts {  
    @Scripts.Render("~/bundles/jqueryval")
```

```
}
```

### **Dashboard.cshtml:**

```
@model customer.Models.custinfo
```

```
@{  
    ViewBag.Title = "Dashboard";  
    Layout = "~/Views/Shared/_Layout.cshtml";  
}
```

```
<h2>Dashboard</h2>
```

```
@using (Html.BeginForm())
```

```
{
```

```
    @Html.AntiForgeryToken()
```

```
    <div class="form-horizontal">
```

```
        <h4>custinfo</h4>
```

```
        <hr />
```

```
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
```

```
        <div class="form-group">
```

```
            @Html.LabelFor(model => model.logid, htmlAttributes: new { @class =  
"control-label col-md-2" })
```

```
            <div class="col-md-10">
```

```
                @Html.EditorFor(model => model.logid, new { htmlAttributes = new {  
@class = "form-control" } })
```

```
                @Html.ValidationMessageFor(model => model.logid, "", new { @class  
= "text-danger" })
```

```
            </div>
```

```
        </div>
```

```
        <div class="form-group">
```

```
            @Html.LabelFor(model => model.customer_email, htmlAttributes: new {  
@class = "control-label col-md-2" })
```

```
            <div class="col-md-10">
```

```

        @Html.EditorFor(model => model.customer_email, new {
htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.customer_email, "", new
{ @class = "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.customer_name, htmlAttributes: new {
@class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.customer_name, new {
htmlAttributes = new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.customer_name, "", new
{ @class = "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.log_status, htmlAttributes: new {
@class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.log_status, new { htmlAttributes =
new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.log_status, "", new {
@class = "text-danger" })
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.descriptions, htmlAttributes: new {
@class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.descriptions, new { htmlAttributes =
new { @class = "form-control" } })
        @Html.ValidationMessageFor(model => model.descriptions, "", new {
@class = "text-danger" })
    </div>
</div>

```



```

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Create" class="btn btn-default" />
            </div>
        </div>
    </div>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

Layout:
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - My ASP.NET Application</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
</head>
<body>
    <div class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse"
data-target=".navbar-collapse" title="more options">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                @Html.ActionLink("Custom Log", "Index", "Home", new { area = "" },
new { @class = "navbar-brand" })
            </div>
            <div class="navbar-collapse collapse">
                <ul class="nav navbar-nav">

```

```
<li>@Html.ActionLink("Home", "Index", "Home")</li>
<li>@Html.ActionLink("About", "About", "Home")</li>
<li>@Html.ActionLink("Contact", "Contact", "Home")</li>
</ul>
</div>
</div>
</div>
<div class="container body-content">
    @RenderBody()
    <hr />
    <footer>
        <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
    </footer>
</div>

@Scripts.Render("~/bundles/jquery")
@Scripts.Render("~/bundles/bootstrap")
@RenderSection("scripts", required: false)
</body>
</html>
```