

一、项目功能和项目介绍

本程序的项目功能在于建立一个能使得“秦时明月”动漫粉丝之间产生更多互动的网站。吸引更多人对这部动漫感兴趣，让他们更加了解动漫，从中获得乐趣。

项目介绍：本网站是一个即创建角色（用户注册登录）、做有关“秦时明月”的趣味测试题、获取“秦迷”专属称号、查阅编辑“秦时角色”等功能于一体的网站。实现了用户与网站，网站与数据库之间的交互，让用户具有身临其境的体验。

二、需求分析

本程序面向的客户群体为：“秦时明月”公司与“秦时明月”的粉丝。

“秦时明月”公司可以通过本程序增大其公众知名度，进而提高收益、效益。

“秦时明月”粉丝可以通过本程序实现虚拟互动，满足了动漫爱好者们的互动需求。

三、页面设计和说明

菜单即展示了页面本布与个数。



主要由以下页面组成：（后文将会有详细的论述）

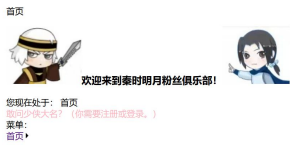
首页（Home）、注册（register）、登录（Login）、娱乐互动（entertainment）、测试题（exam）、角色（Roles）、称号（Mytitle）

四、使用说明与测试结果

本程序实现功能如下：注册；登录；写测试题；查看编辑人物角色信息；查询该用户最近一次测试题所得称号。

1.注册

首先进入“首页”页面。



将鼠标光标移动至菜单下的首页位置将会展示整体菜单。

菜单:

首页▶注册

登录

娱乐互动▶来场秦时测试题?

秦时角色一览(查看编辑)

我的称号

点击“注册”即可进入注册界面。后续的功能进入方式相同。

在注册页面按照要求进行用户名的注册。(若输入不符合要求将会有红色信息提示，直到输入符合要求后，才可提交成功)请记住用户名和密码，将在后续的“登录界面”中使用。

下左图为刚进入页面，右图为未满足要求而提交失败的页面。查看并确认好填入的信息后，点击“确认”，系统将会显示提交成功。

此时，可以用该账号登录系统了。

秦时明月用户注册页面



欢迎来到秦时明月粉丝俱乐部!

您现在处于: [首页](#) > 注册

敢问少侠大名? (你需要注册或登录。)

菜单:

首页▶

秦时明月用户注册页面



欢迎来到秦时明月粉丝俱乐部!

您现在处于: [首页](#) > 注册

敢问少侠大名? (你需要注册或登录。)

菜单:

首页▶

您好, 新秦迷!

欢迎进入秦时明月用户注册界面!

*姓名:

*设置密码: *两次密码输入不一致!

*确认密码:

*性别:
☐男 ☐女

生日:

*出生地:

*联系电话:

版权信息:Copyright © 2020 ,
联系电话:+8
沪安网安备:3

您好, 新秦迷!

欢迎进入秦时明月用户注册界面!

*姓名: *姓名不能为空!

*设置密码: *两次密码输入不一致!

*确认密码:

*性别:
☐男 ☐女
*请选择性别!

生日:

*出生地: *出生地不能为空!

*联系电话: *联系电话不能为空!

版权信息:Copyright © 2020 A
联系电话:+8
沪安网安备:3

请确认信息

姓名:
密码: 123456
性别: 女
生日: 2001/02/28
出生地: 南昌
联系电话: 131

提交成功!

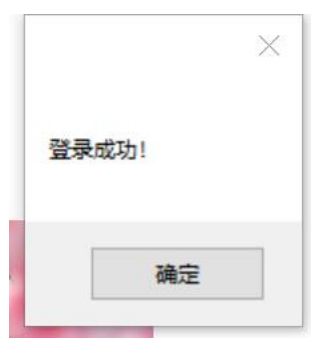
2.登录

注册成功后，用图一的方法，从菜单点击“登录”进入登录页面。此处有一个小细节，当前位置会随着进入的页面产生变化，箭头指向的最后一个为当前位置。

若用户名不存在或用户名与密码不匹配，将出现左图提示语；反之，成功登录为右边提示语，登录成功后密码输入框底部的“欢迎来到秦时明月！”将会被替换成“秦迷+【用户名】，欢迎回来~我们等你很久了！”。

刷新或进入其他页面，粉色字体的“敢问少侠大名？（你需要注册或登录。）”将会被替换成“亲爱的秦迷+【用户名】~现在你想做些什么~？”此刻完成用户登录功能，可进入娱乐互动页面。

秦时明月登录页面



秦时明月登录页面



亲爱的秦迷[黑]钰~现在你想做些什么~?

3.写测试题

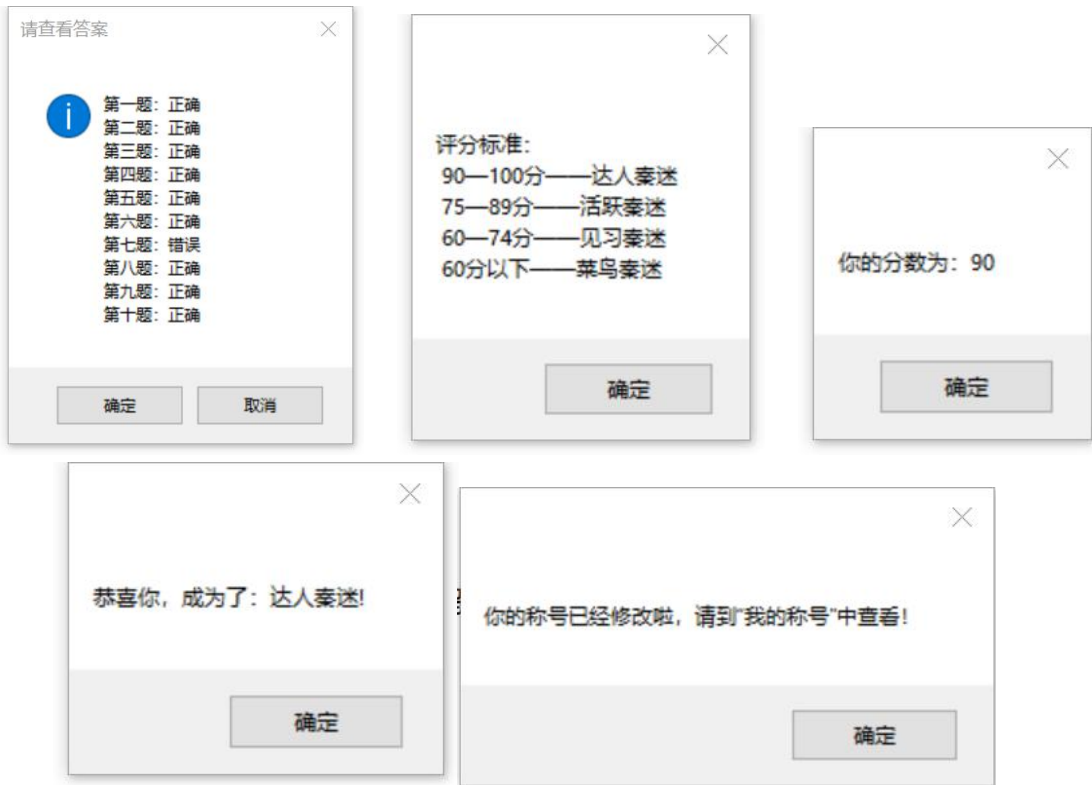
娱乐互动页面如下图所示。可以注意到当前位置页的箭头文字也发生了变化。



从菜单点击“来场秦时测试题？”进入测试题页面。答题即可。



答题完毕后，点击“提交答案”按钮出现连续如下提示。



可以多次答题，称号以最后一次结果为准。

答题完毕即可在“我的称号”中获取当前用户的当前称号。

4. 查询当前用户当前称号

如果用户在新注册并登录系统后，从未参加过测试，则在查询称号时显示“暂无”。用户点击“点击我！”按钮，则橙色字体“点击获取按钮处”将会变成用户称号。重新参加测试并获得新称号，则会更新。

我的称号



欢迎来到秦时明月粉丝俱乐部!



您现在处于: [首页](#) > [娱乐互动](#) > 我的称号

亲爱的秦迷 [REDACTED] 哇~现在你想做些什么~?

菜单:

[首页](#) ▾

你目前的称号为: [点击按钮获取称号](#)

[请点我!](#)

版权信息: Copyright © 2020 A

联系电话: +8

沪安网安备: 3

语句，函数最后返回一个字符串。

```
1 个引用
protected string getSelectString()
//将用户填写的姓名，密码拼凑成查询语句
{
    string msg1 = TextBox1.Text.ToString();
    string msg2 = TextBox2.Text.ToString();
    return "select 识别字符串 from 用户 where 姓名='" + msg1 + "' and 密码='" + msg2 + "'";
}
```

第二个函数的功能在于连接数据库，通过 reader 得到值，再做出登录成功与否的判断。

```
0 个引用
protected void Button1_Click(object sender, EventArgs e)
//登录按钮点击后，将第一个文本框与第二个文本框的值送入查询语句中，
//若在数据库中查询出相应的行(HasRows)，则表示用户存在且密码正确，否则登录失败
{
    string cs = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=d:\Users\LJY\source\r
        Security=True";

    SqlConnection connection = new SqlConnection(cs);
    SqlCommand command = connection.CreateCommand();
    command.CommandText = getSelectString();

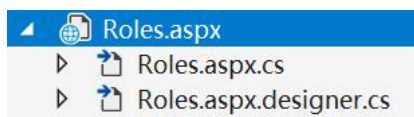
    try
    {
        connection.Open();
        SqlDataReader reader = command.ExecuteReader();
        reader.Read();

        if (reader.HasRows)
        //由于数据库中设置了标志字符串，所以有查询结果即表示用户存在
        {
            string text = reader[0].ToString();
            MessageBox.Show("登录成功!");
            Label1.Text = "秦迷" + TextBox1.Text.ToString() + ", 欢迎回来~我们等你很久了!";
            Session["username"] = TextBox1.Text.ToString();
        }
        else{
            //没有结果表示用户名不存在或密码错误
            MessageBox.Show("用户名不存在或密码不正确，请重试或注册!");
        }
    }
    catch { }
    finally
    {
        connection.Close();
    }
}
```

值得一提的是，Label1 的变化效果与母版页中 Label 的变化效果类似，原理相近。区别在于母版页的 Label 不会因为从菜单进入到其他页面而消失，登录页面的 label 只在本页面登陆成功时发生变化。

2.数据库的连接与查询语句（增、删、改）

实现数据库与页面的交互，是页面程序设计的重点。本程序主要采取两种方式达到目的，第一种是 Gridview 的数据绑定，这种方法的好处在于代码的编写较少（但也需要研究功能，且不易实现“增”这一操作），在“秦时角色一览”（即 Roles 窗体）中，采用的是该方法。



该方法还需注意的问题是表定义中的非空项，应在存储路径（第二张图）中删去相应的非空项（该项不打算编辑的情况下，如原本打算做成编辑，则不需要删去。），否则会有报错表示不可使非空项的值为 NULL。

```
<EditItemTemplate><b></b>
<asp:Label ID="Label1" runat="server" Text='<%# Bind("角色ID") %>' /> -
<asp:DropDownList id="dropdownlist1" runat="server"
    DataSource="<%# CourtesyTitle %>"
    SelectedValue='<%# Bind("身份") %>' AutoPostBack="False">
</asp:DropDownList>
<%# Eval ("姓氏") %><%# Eval ("名字") %><br /></b>
<%# Eval ("年龄") %><br />
<asp:TextBox ID="TextBox5" runat="server"
    Text='<%# Bind("年龄") %>'
    TextMode="Multiline"></asp:TextBox><br />
<asp:TextBox ID="TextBox1" runat="server"
    Text='<%# Bind("备注") %>'
    TextMode="Multiline"></asp:TextBox><br />
</EditItemTemplate>
```

上图可编辑的属性为“身份”（下拉选择），“年龄”（文本框），“备注”（文本框）。

```
<asp:SqlDataSource ID="SqlDataSource_Roles" runat="server" ConnectionString="<%$ ConnectionStrings:ConnectionString %>" DeleteCommand="DELETE FROM [秦时角色] WHERE [角色ID] = @角色ID" InsertCommand="INSERT INTO [秦时角色] ([姓氏], [名字], [身份], [年龄], [照片], [备注]) VALUES (@姓氏, @名字, @身份, @年龄, @照片, @备注)" SelectCommand="SELECT * FROM [秦时角色]" UpdateCommand="UPDATE [秦时角色] SET [身份] = @身份, [年龄] = @年龄, [备注] = @备注 WHERE [角色ID] = @角色ID">
```

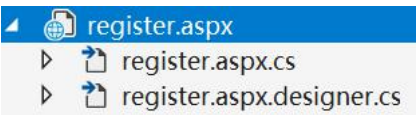
上图可见 Update 中删去了“姓氏”、“名字”（因为两值非空且没有作为可编辑项），表定义见下图。

输出	dbo.秦时角色 [设计]	脚本文件(I):	dbo.秦时角色.sql
名称	数据类型	允许 Null	默认值
角色ID	int	<input type="checkbox"/>	
姓氏	nvarchar(20)	<input type="checkbox"/>	
名字	nvarchar(10)	<input type="checkbox"/>	
身份	nvarchar(30)	<input checked="" type="checkbox"/>	
年龄	nvarchar(10)	<input checked="" type="checkbox"/>	
照片	nvarchar(255)	<input checked="" type="checkbox"/>	
备注	nvarchar(MAX)	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

第二种与数据库交互的方法为建立与数据库接口连接，再通过调用 command，改变 command 的 text（用数据库的查询语句）来实现交互。

笔者采用这种方式的主要思想在于：首先，编写一个专门用于将用户信息（例如来自文本框）提取出来，再加上相应的查询语句（如，select，update，insert 等）整合到一个字符串中，最后返回它。在赋值给 command 的 text 时，直接调用该函数即可达到目的。

以“注册窗体”（即 register 窗体）为例：



Register.aspx 中的验证控件、正则表达式不再赘述，它的功能在于通过提交按钮得到文本框内数据，直接看 cs 文件中的函数。

下图一的作用在于合成一个具有“确认输入信息”的字符串，方便输出。

下图二的作用在于合成一个“插入语句”的字符串。值得注意的地方是称号的“暂无”。

```

1 个引用
protected string getValue()
//将六个信息整合并加入换行符成为一个字符串
{
    string msg1 = "姓名: " + TextBox1.Text + "\n";
    string msg2 = "密码: " + TextBox5.Text + "\n";
    string msg3 = "性别: " + RadioButtonList1.SelectedValue + "\n";
    string msg4 = "生日: " + TextBox2.Text + "\n";
    string msg5 = "出生地: " + TextBox3.Text + "\n";
    string msg6 = "联系电话: " + TextBox4.Text + "\n";

    return msg1 + msg2 + msg3 + msg4 + msg5 + msg6;
}

1 个引用
protected string getSqlString()
//将用户填写的数据的字符串拼接起来成为一个插入查询语句
{
    string msg1 = TextBox1.Text.ToString();
    string msg2 = TextBox5.Text.ToString();
    string msg3 = RadioButtonList1.SelectedValue.ToString();
    string msg4 = TextBox2.Text;
    string msg5 = TextBox3.Text.ToString();
    string msg6 = TextBox4.Text;
    string msg7 = msg1 + msg2;
    return "insert into 用户 Values('"+ msg1 + "','"+ msg2 + "','"+ msg3 + "','"+ msg4 + "','"+ msg5 + "','"+ msg6 + "','"+ msg7 + "','暂无')";
}

```

第三张图函数的作用在于连接数据库，实现插入新用户至用户表的操作（不用从中读取数据）。

```

protected void Button1_Click(object sender, EventArgs e)
//确认提示框，第一个显示内容，第二个提示框标题
{
    DialogResult result = MessageBox.Show(getValue(), "请确认信息", MessageBoxButtons.OKCancel, MessageBoxIcon.Information);

    string cs = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=d:\Users\LJY\source\repos\WebApplicationFinal_LJY\App_Data\Northwind.mdf;Integrated Security=True";

    SqlConnection connection = new SqlConnection(cs);
    SqlCommand command = connection.CreateCommand();
    command.CommandText = getSqlString();
    try
    {
        connection.Open();
        command.ExecuteNonQuery();
    }
    catch { }
    finally
    {
        connection.Close();
    }
    MessageBox.Show("提交成功!");
}

```

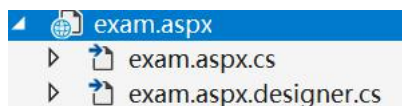
以上便是第二种方法与数据库交互的基本实现思路，在之前的窗体中大都运用该类方法。

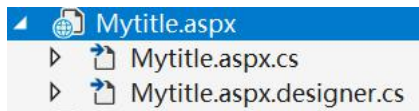
3.Session 的妙用

Session 的巧妙之处在于储存当前用户的姓名，类似于全局变量，这点非常实用，尤其在实现母版页的“显示当前用户”功能上非常实用。在登录（Login）窗体中，一旦匹配成功，将会把当前用户名放入 Session 中储存，当再次刷新页面时，会触发母版页中的 load 函数中的代码（前有截图提及），从而实现上述功能。

4.测试题与称号

实现测试题与称号的两窗体如下。





测试题的 cs 文件整合字符串方式与前相似，不再单独列出。主要式判断题目正确与否的函数与算分函数，难度不大，重点在于字符串与数字的转换。

```
1 个引用
protected string GetValue()
//使用convert把得到的value由字符串转换为int, 判断是否为0, 若为0, 则表示选项错误, 1表示选项正确
{
    string msg1 = "第一题: " + ((Convert.ToInt32(RadioButtonList1.SelectedValues) == 0) ? "错误" : "正确") + "\n";
    string msg2 = "第二题: " + ((Convert.ToInt32(RadioButtonList2.SelectedValues) == 0) ? "错误" : "正确") + "\n";
    string msg3 = "第三题: " + ((Convert.ToInt32(RadioButtonList3.SelectedValues) == 0) ? "错误" : "正确") + "\n";
    string msg4 = "第四题: " + ((Convert.ToInt32(RadioButtonList4.SelectedValues) == 0) ? "错误" : "正确") + "\n";
    string msg5 = "第五题: " + ((Convert.ToInt32(RadioButtonList5.SelectedValues) == 0) ? "错误" : "正确") + "\n";
    string msg6 = "第六题: " + ((Convert.ToInt32(RadioButtonList6.SelectedValues) == 0) ? "错误" : "正确") + "\n";
    string msg7 = "第七题: " + ((Convert.ToInt32(RadioButtonList7.SelectedValues) == 0) ? "错误" : "正确") + "\n";
    string msg8 = "第八题: " + ((Convert.ToInt32(RadioButtonList8.SelectedValues) == 0) ? "错误" : "正确") + "\n";
    string msg9 = "第九题: " + ((Convert.ToInt32(RadioButtonList9.SelectedValues) == 0) ? "错误" : "正确") + "\n";
    string msg10 = "第十题: " + ((Convert.ToInt32(RadioButtonList10.SelectedValues) == 0) ? "错误" : "正确") + "\n";

    return msg1 + msg2 + msg3 + msg4 + msg5 + msg6 + msg7 + msg8 + msg9 + msg10;
}

3 个引用
protected string Total_score()
//使用convert把得到的value由字符串转换为int, 后计算值*20
{
    int score = 0;
    score = 10 * (Convert.ToInt32(RadioButtonList1.SelectedValues) + Convert.ToInt32(RadioButtonList2.SelectedValues) + Convert.ToInt32(RadioButtonList3.SelectedValues) + Convert.ToInt32(RadioButtonList4.SelectedValues) + Convert.ToInt32(RadioButtonList5.SelectedValues) + Convert.ToInt32(RadioButtonList6.SelectedValues) + Convert.ToInt32(RadioButtonList7.SelectedValues) + Convert.ToInt32(RadioButtonList8.SelectedValues) + Convert.ToInt32(RadioButtonList9.SelectedValues) + Convert.ToInt32(RadioButtonList10.SelectedValues));
    return score.ToString();
}
```

称号采用的也是类似于前面的拼接字符串、查询数据库操作。

```
1 个引用
protected string getSelectString_mytitle()
//将用户填写的姓名拼凑成查询语句
{
    string msg1 = Session["username"].ToString();

    return "select 称号 from 用户 where 姓名='" + msg1 + "'";
}

0 个引用
protected void Button1_Click(object sender, EventArgs e)
{
    string cs = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=d:\Users\LJY\source\repos\WebApplicationFinal_LJY\App_Data\Northwind.mdf;Integrated Security=True";

    SqlConnection connection = new SqlConnection(cs);
    SqlCommand command = connection.CreateCommand();
    command.CommandText = getSelectString_mytitle();
    try
    {
        connection.Open();
        SqlDataReader reader = command.ExecuteReader();
        reader.Read();
        Label1.Text = reader[0].ToString();
    }
    catch { }
    finally
    {
        connection.Close();
    }
}
```

5. 站点地图、菜单、当前路径

在站点地图中写好对应的逻辑树结构，之后只要在工具箱中导航栏中找到菜单（Menu）、当前路径（Path）导入站点地图作为数据源就可实现功能了。站点地图代码如下：

笔者在这方面遇到的困难主要在于排版的美观，发现引入 div 是一个比较好的方法。


```

<siteMapNode url="Home.aspx" title="首页" description="">
  <siteMapNode url="register.aspx" title="注册" description="" >
    </siteMapNode>

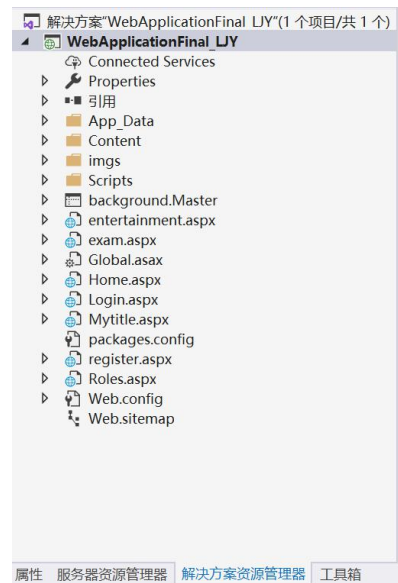
    <siteMapNode url="Login.aspx" title="登录" description="" >
      </siteMapNode>

    <siteMapNode url="entertainment.aspx" title="娱乐互动" description="" >
      <siteMapNode url="exam.aspx" title="来场秦时测试题?" description="" />
      <siteMapNode url="Roles.aspx" title="秦时角色一览（查看编辑）" description="" />
      <siteMapNode url="Mytitle.aspx" title="我的称号" description="" />
    </siteMapNode>
  </siteMapNode>
</siteMapNode>

```

六、结论

项目全部文件截图如下：



其实还可以添加很多功能，网站的框架已经搭建起来了，比如“统计在线人数”、“统计访问次数”、“零点关闭服务器”等等，只要实现与数据库的交互，这些功能的增加与完善就是一件很顺畅的事情。整体来说，经过这些天的编程，达到了一个还算满意的效果，为自己感兴趣的事物编程还是很有趣的，之后会试着再增加些功能，使这个程序更完善，功能更齐全一些。