**《ASP.NET程序设计实验》课程设计**

**题目： 任务展示管理系统-TaskManagePro**

**专业班级**  信计1701班

**学生姓名**  xxx

**指导教师** xxx

**提交日期** 2020.6.02

目录

**[一、 设计目的](#_Toc2076696568_WPSOffice_Level1)** **[2](#_Toc2076696568_WPSOffice_Level1)**

[1.1 绪论](#_Toc887070451_WPSOffice_Level2) [2](#_Toc887070451_WPSOffice_Level2)

**[二、 设计要求和设计指标](#_Toc887070451_WPSOffice_Level1)** **[2](#_Toc887070451_WPSOffice_Level1)**

[1. 系统基本要求](#_Toc111026904_WPSOffice_Level2) [2](#_Toc111026904_WPSOffice_Level2)

[2. 系统功能要求](#_Toc2013369932_WPSOffice_Level2) [3](#_Toc2013369932_WPSOffice_Level2)

**[三、 设计内容](#_Toc1161592483_WPSOffice_Level1)** **[3](#_Toc1161592483_WPSOffice_Level1)**

[3.1 技术栈](#_Toc808621345_WPSOffice_Level2) [3](#_Toc808621345_WPSOffice_Level2)

[3.2 整体设计&系统层次](#_Toc1222427199_WPSOffice_Level2) [3](#_Toc1222427199_WPSOffice_Level2)

[3.3 具体设计](#_Toc357882744_WPSOffice_Level2) [4](#_Toc357882744_WPSOffice_Level2)

[3.4 系统运行效果&测试](#_Toc1203378968_WPSOffice_Level2) [19](#_Toc1203378968_WPSOffice_Level2)

**[四、 本设计改进建议](#_Toc111026904_WPSOffice_Level1)** **[23](#_Toc111026904_WPSOffice_Level1)**

**[五、 总结](#_Toc2013369932_WPSOffice_Level1)** **[23](#_Toc2013369932_WPSOffice_Level1)**

**[六、 主要参考文献](#_Toc808621345_WPSOffice_Level1)** **[24](#_Toc808621345_WPSOffice_Level1)**

**[附录](#_Toc1203378968_WPSOffice_Level1)** **[24](#_Toc1203378968_WPSOffice_Level1)**

1. **设计目的**

**1.1 绪论**

随着网络技术的飞速发展，互联网已经成为人们生活中 不可缺少的一部分。在社交中，互联网已经成为了当今最主流的社交方式，人们通过互联网上传大量的各类资源来进行共享，如何对这些资源进行分类和展示，成为了一个待解决的问题。

笔者经过<<ASP.NET程序设计>>这门课程的学习，掌握了ASP.NET应用网站的基本开发知识，为解决这个问题打下了开发的基础。在本课程的学习过程中，同学们单独申请了一些ECS云服务器，用于在云上运行自己的应用程序，但是由于缺乏一个统一的展示平台，导致同学们网站建设知识不能够很好及时地进行内部共享，只能通过聊天软件等方式来互相交流，如果网站有更新也不能及时地告知。

基于上述问题，一款基于ASP.NET框架的任务展示管理系统TaskManagerPro便是笔者对该系列问题的一个解答。

1. **设计要求和设计指标**

任务展示管理系统TaskManagerPro应该具备基础的用户注册、用户登录、修改用户信息、添加展示内容、修改展示内容、管理展示内容等基本功能，同时在前端设计上面考虑足够的兼容性，保持移动端和web端的完美展示和体验，也具备良好的程序健壮性、合理的程序架构，清晰的业务逻辑等基本特点，能够实现在云端轻松部署，开箱即用。

**相关参数指标:**

1. **系统基本要求**
2. 系统后端基于MVC三层架构搭建。
3. 采用Linq技术。
4. 应用母版技术。
5. UI界面呈现方式是WEB界面，应用强大CSS与Javascript技术。
6. 采用SQLSERVER数据库。
7. **系统功能要求**
8. 用户注册、用户登录
9. 修改密码、找回密码、个人信息编辑
10. ECS信息管理(编辑和控制)
11. 公告板内容实时刷新
12. 网站状态功能展示
13. **设计内容**

登录、注册、修改密码、找回密码、编辑个人信息、编辑个人的ECS信息、ECS展示功能控制、公告板内容动态刷新。

**3.1 技术栈**

1. 开发工具：visual stdio 2017
2. 后端: MVC三层结构
3. 数据库： mysql
4. 持久层： Linq
5. 前端： css html js Bootstrap框架(核心)
6. 开发部署方式：本地部署/云端部署
7. 作图工具：oneNote、proccesson

**3.2 整体设计&系统层次**

**根据设计内容进行功能模块的划分**

(1) 用户功能

(2) 任务展示功能

(3) 网站实时状态

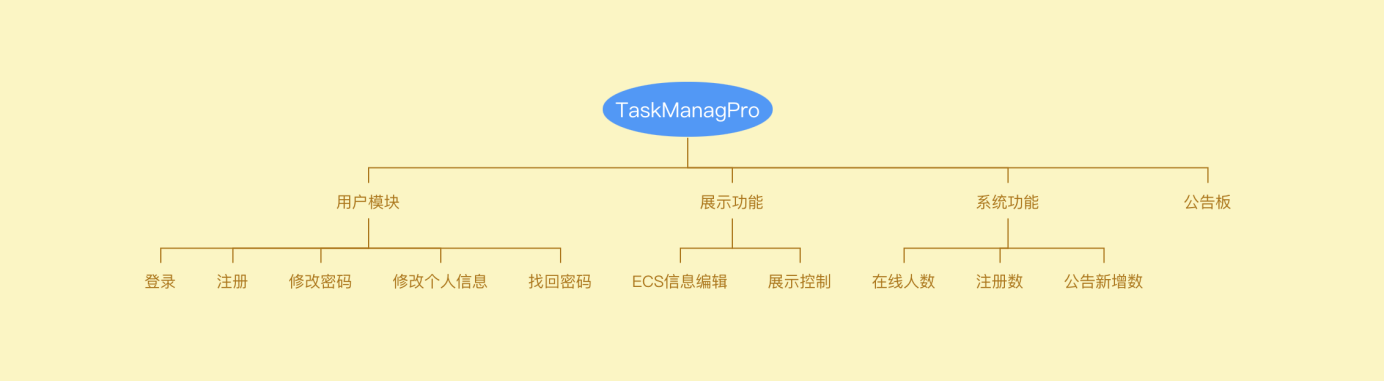


图 3.2.1系统整体架构图

**3.3 具体设计**

**数据库设计:**

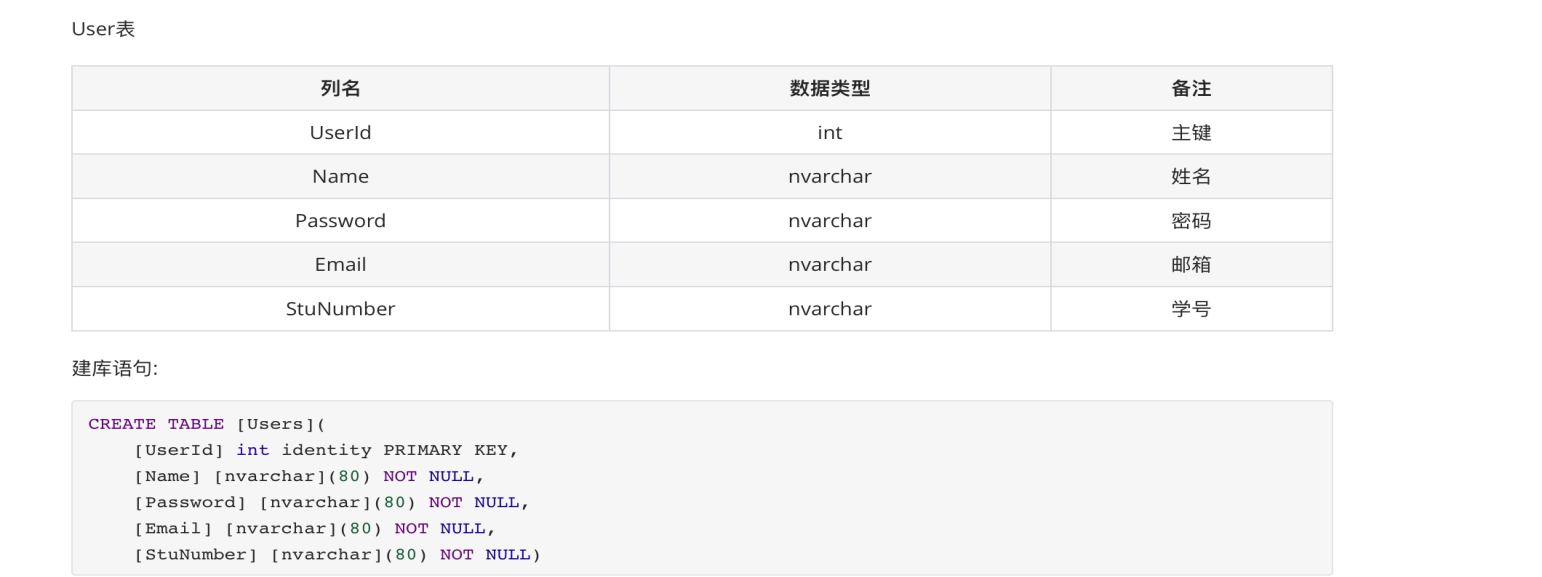


图 3.3.1 User表



图 3.3.2 Person表

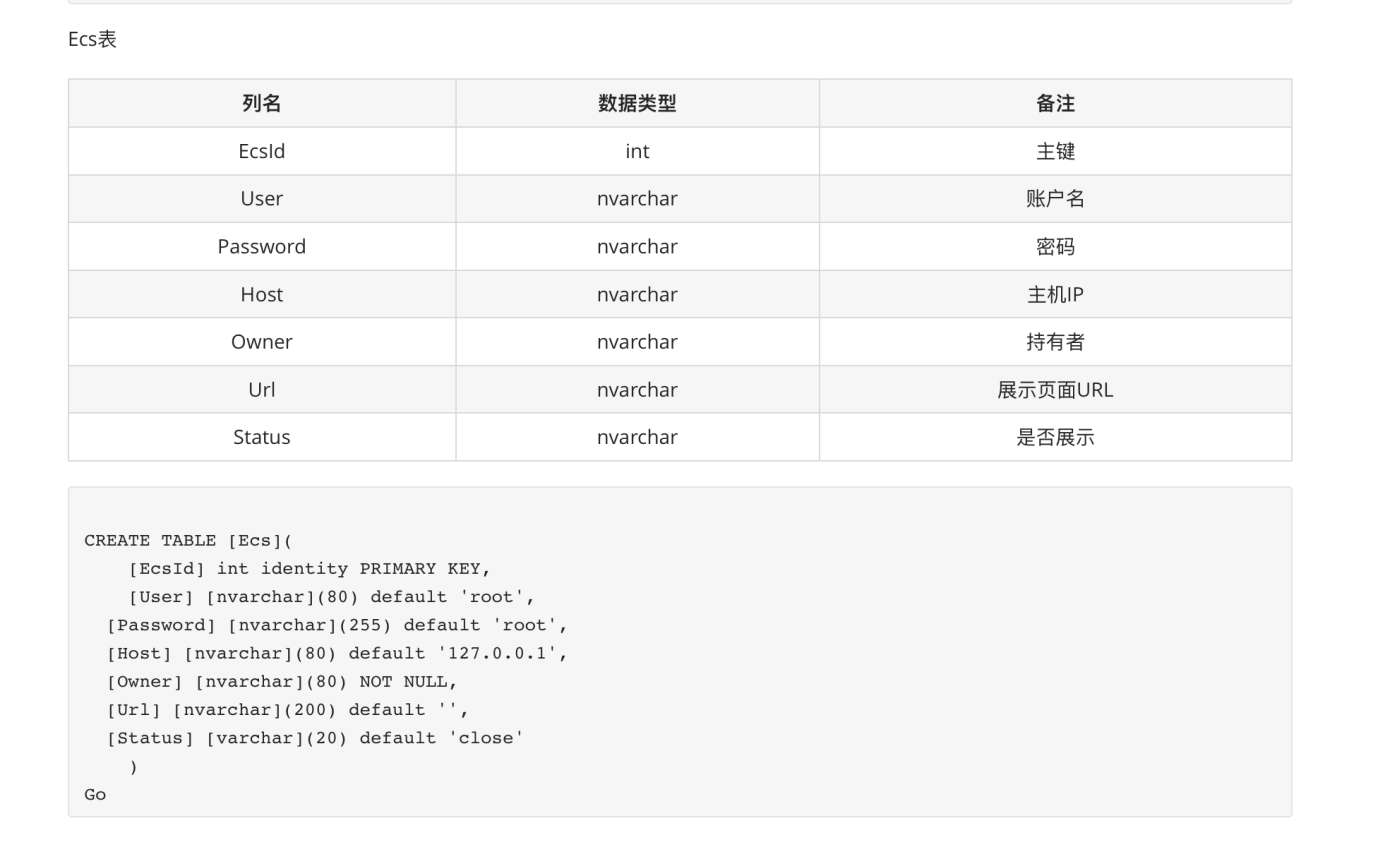


图 3.3.3 Ecs表

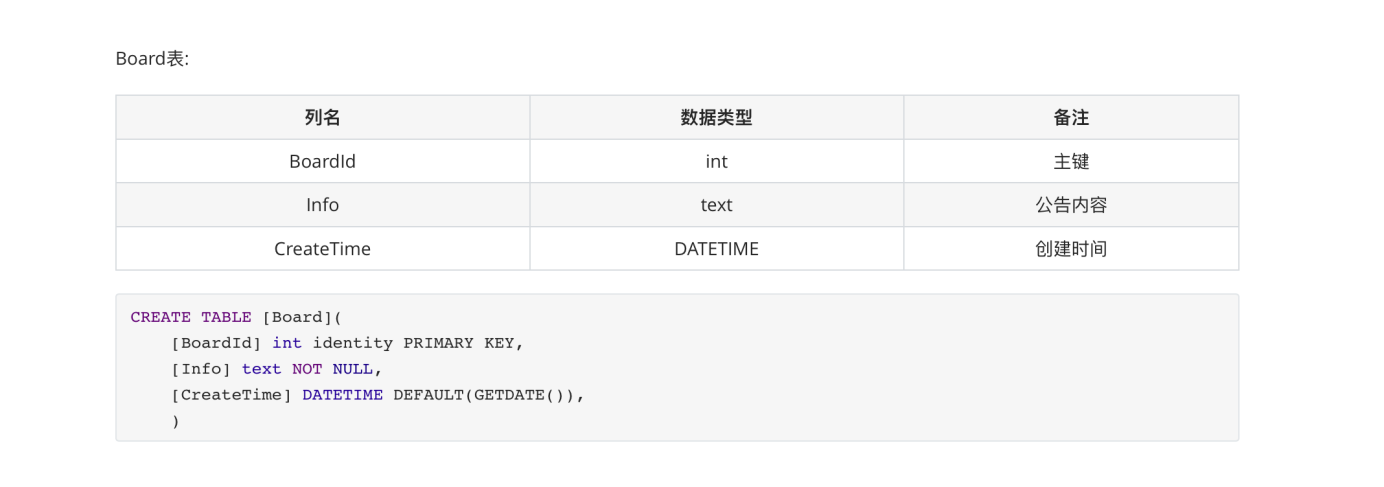


图 3.3.4 Board表

部分代码:

**BoardService类**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using TaskManagePro.DAL;

namespace TaskManagePro.BLL

{

public class BoardService

{

TaskManageProDataContext db = new TaskManageProDataContext();

public void Insert(String info)

{

Board board = new Board

{

Info = info,

};

db.Board.InsertOnSubmit(board);

db.SubmitChanges();

}

public List<Board> GetAllInfo(int n)

{

return (from c in db.Board orderby c.BoardId descending

select c).Take(n).ToList();

}

public int GetCount()

{

return (from c in db.Board

select c).Count();

}

}

}

**EcsService类**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using TaskManagePro.DAL;

namespace TaskManagePro.BLL

{

public class EcsService

{

TaskManageProDataContext db = new TaskManageProDataContext();

public void Insert(string owner)

{

Ecs ecs = new Ecs

{

Owner = owner,

};

db.Ecs.InsertOnSubmit(ecs);

db.SubmitChanges();

}

public Ecs SelectEcs(string owner)

{

Ecs ecs = (from c in db.Ecs

where c.Owner == owner

select c).First();

return ecs;

}

public void UpdateEcs(string user, string password, string host, string url, string owner)

{

Ecs ecs = (from c in db.Ecs

where c.Owner == owner

select c).First();

ecs.User = user;

ecs.Password = password;

ecs.Host = host;

ecs.Url = url;

db.SubmitChanges();

}

public void UpdateStatus(string status, string owner)

{

Ecs ecs = (from c in db.Ecs

where c.Owner == owner

select c).First();

ecs.Status = status;

db.SubmitChanges();

}

public bool IsNameExist(string owner)

{

Ecs ecs = (from c in db.Ecs

where c.Owner == owner

select c).FirstOrDefault();

if (ecs != null)

{

return true;

}

else

{

return false;

}

}

}

}

**PersonService类**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using TaskManagePro.DAL;

namespace TaskManagePro.BLL

{

public class PersonService

{

TaskManageProDataContext db = new TaskManageProDataContext();

public void Insert(string name, string nickname, string email, string stunumber, string introduce, string sex)

{

Person person = new Person

{

Name = name,

NickName = nickname,

Email = email,

StuNumber = stunumber,

Introduce = introduce,

Sex = sex

};

db.Person.InsertOnSubmit(person);

db.SubmitChanges();

}

public void UpdatePerson(string nickname, string email, string introduce, string sex, string name)

{

Person person = (from c in db.Person

where c.Name == name

select c).First();

person.NickName = nickname;

person.Email = email;

person.Introduce = introduce;

person.Sex = sex;

db.SubmitChanges();

}

public Person SelectPerson(string name)

{

Person person = (from c in db.Person

where c.Name == name

select c).First();

return person;

}

}

}

**UserService类**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using TaskManagePro.DAL;

namespace TaskManagePro.BLL

{

public class UserService

{

TaskManageProDataContext db = new TaskManageProDataContext();

// 检查登录

public int CheckLogin(string name, string password)

{

Users customer = (from c in db.Users

where c.Name == name && c.Password == password

select c).FirstOrDefault();

if (customer != null) //用户名和密码正确

{

return customer.UserId;

}

else //用户名或密码错误

{

return 0;

}

}

public bool IsNameExist(string name)

{

Users customer = (from c in db.Users

where c.Name == name

select c).FirstOrDefault();

if (customer != null)

{

return true;

}

else

{

return false;

}

}

public bool IsNoExist(string no)

{

Users customer = (from c in db.Users

where c.StuNumber == no

select c).FirstOrDefault();

if (customer != null)

{

return true;

}

else

{

return false;

}

}

public void Insert(string name, string password, string email,string no)

{

Users customer = new Users

{

Name = name,

Password = password,

Email = email,

StuNumber = no

};

db.Users.InsertOnSubmit(customer);

db.SubmitChanges();

}

public void ResetPassword(string name, string email)

{

Users customer = (from c in db.Users

where c.Name == name && c.Email == email

select c).First();

customer.Password = name;

db.SubmitChanges();

}

public void ChangePassword(int customerId, string password)

{

Users customer = (from c in db.Users

where c.UserId == customerId

select c).First();

customer.Password = password;

db.SubmitChanges();

}

public int GetUserCount()

{

int count = (from c in db.Users

select c).Count();

return count;

}

}

}

**Login.aspx.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using TaskManagePro.BLL;

public partial class Login : System.Web.UI.Page

{

UserService customerSrv = new UserService();

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["CustomerName"] != null)

{

Response.Redirect("~/Main.aspx");

}

}

protected void btnLogin\_Click(object sender, EventArgs e)

{

if (true)

{

if (txtName.Text.Trim() == "" || txtPwd.Text.Trim() == "")

{

Response.Write("<script>alert('用户名或密码不能为空!');</script>");

}

else

{

//调用CustomerService类中的CheckLogin()方法检查输入的用户名和密码是否正确

int customerId = customerSrv.CheckLogin(txtName.Text.Trim(), txtPwd.Text.Trim());

if (customerId > 0) //用户名和密码正确

{

Session.Clear(); //清理Session中保存的内容

Session["CustomerId"] = customerId;

Session["CustomerName"] = txtName.Text;

Response.Redirect("~/Main.aspx");

}

else //用户名或密码错误

{

Response.Write("<script>alert('用户名或密码错误!');</script>");

}

}

}

}

}

**Main.aspx.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using TaskManagePro.BLL;

using TaskManagePro.DAL;

public partial class Main : System.Web.UI.Page

{

UserService userService = new UserService();

BoardService boardService = new BoardService();

public int UserCount;

public int OnlineCount;

public int BoardCount;

public string BoardInfo;

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["CustomerName"] == null) //用户未登录

{

Response.Redirect("~/Login.aspx");

}

UserCount = userService.GetUserCount();

OnlineCount = (int)Application["VisitNumber"];

BoardCount = 1;

}

protected void boardTime\_Tick(object sender, EventArgs e)

{

int ItemCount = boardService.GetCount();

BoardCount = ItemCount - (int)Application["BoardCount"];

Application["BoardCount"] = ItemCount;

}

protected void Timer\_\_Tick(object sender, EventArgs e)

{

BoardInfo = "";

List<Board> board = boardService.GetAllInfo(3);

foreach( Board b in board)

{

BoardInfo += b.Info + "<br />";

}

}

}

**ProfileEdit.aspx.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using TaskManagePro.BLL;

using TaskManagePro.DAL;

public partial class ProfileEdit : System.Web.UI.Page

{

PersonService personService = new PersonService();

public string name;

public string nickname;

public string email;

public string stunumber;

public string introduce;

public string sex;

protected void Page\_Load(object sender, EventArgs e)

{

if (Session["CustomerName"] == null) //用户未登录

{

Response.Redirect("~/Login.aspx");

}

name = Session["CustomerName"].ToString();

Person person = personService.SelectPerson(name);

nickname = person.NickName;

email = person.Email;

stunumber = person.StuNumber;

introduce = person.Introduce;

sex = person.Sex;

if(sex == "女")

{

rbFemale.Checked = true;

rbmale.Checked = false;

}

else

{

rbmale.Checked = true;

rbFemale.Checked = false;

}

}

protected void btnChangPwd\_Click(object sender, EventArgs e)

{

}

}

**3.4 系统运行效果&测试**

系统通过提供登录接口，从而进入后台管理中心，在管理中心中，用户可以自由编辑自己的信息，并且应用系统提供的展示功能，进行编辑个人的ECS信息，浏览公告板和网站状态信息，并且决定是否对外公开自己ECS的网站信息，查阅其他用户公开的ECS信息功能。

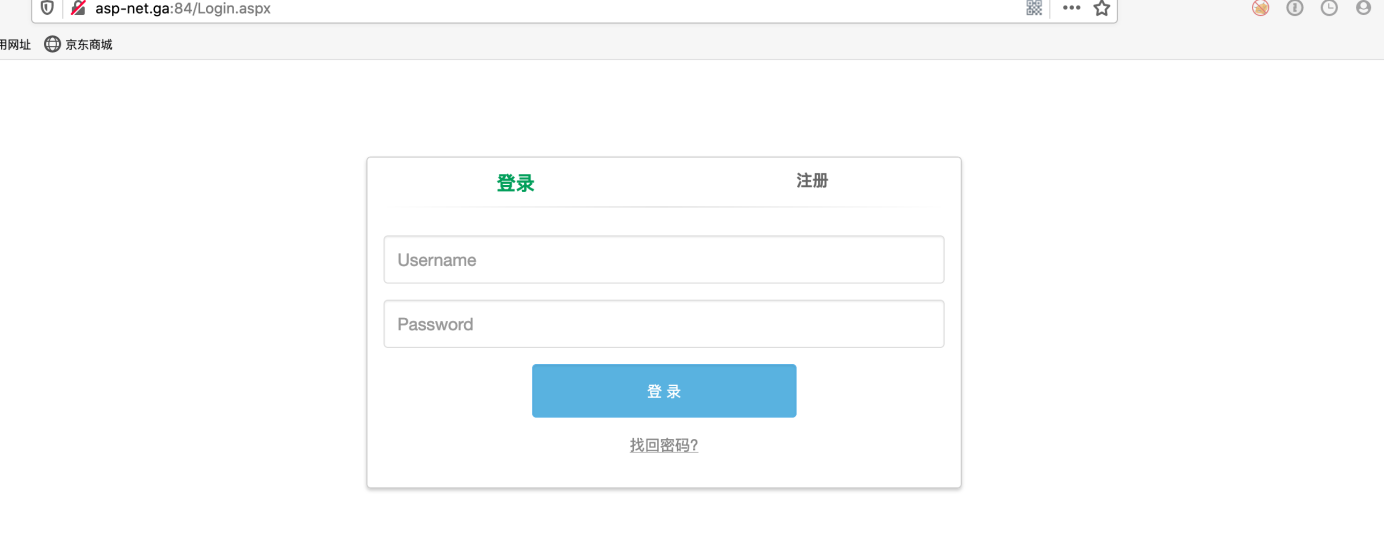


图 3.4.1 登录界面

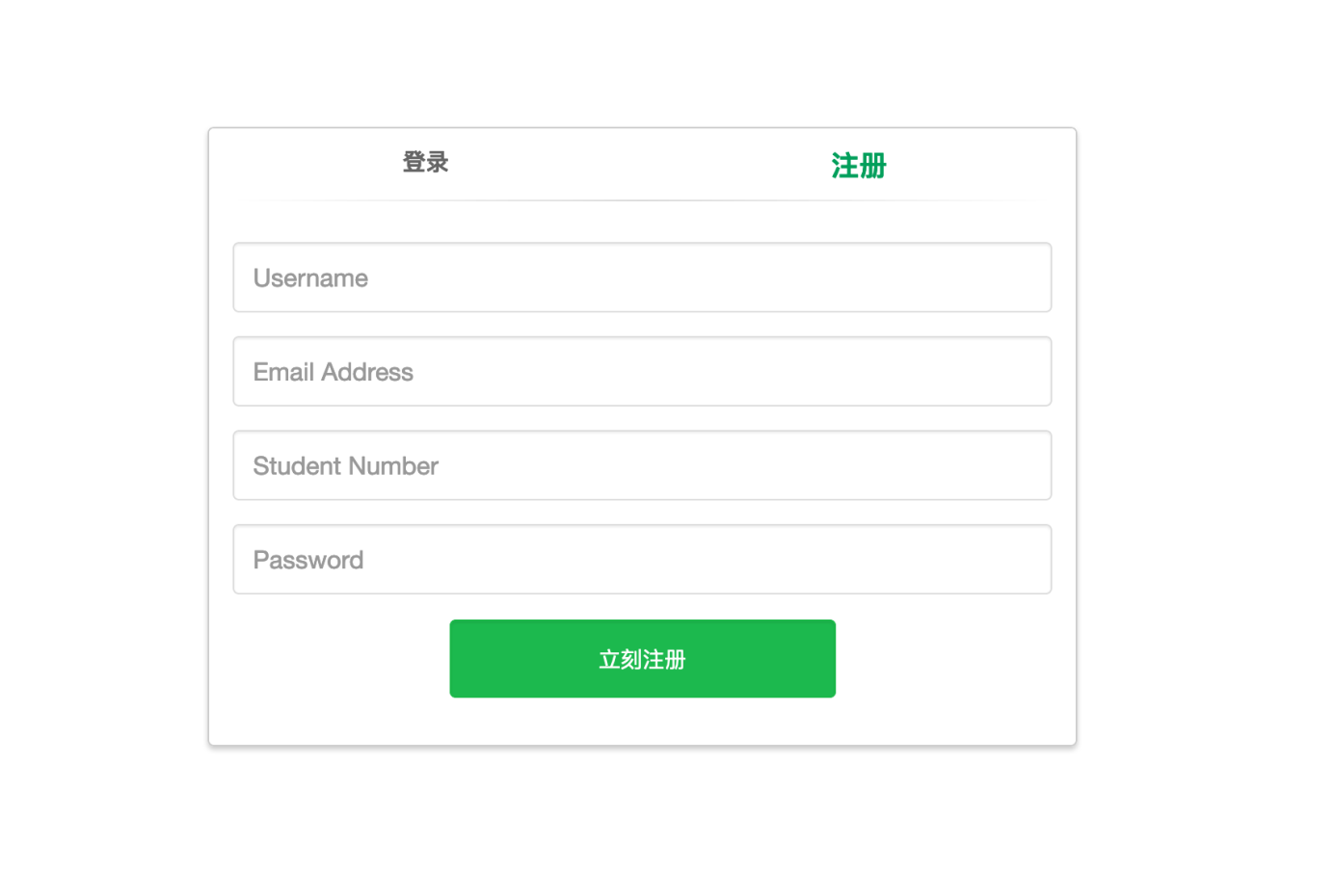


图 3.4.2 注册界面

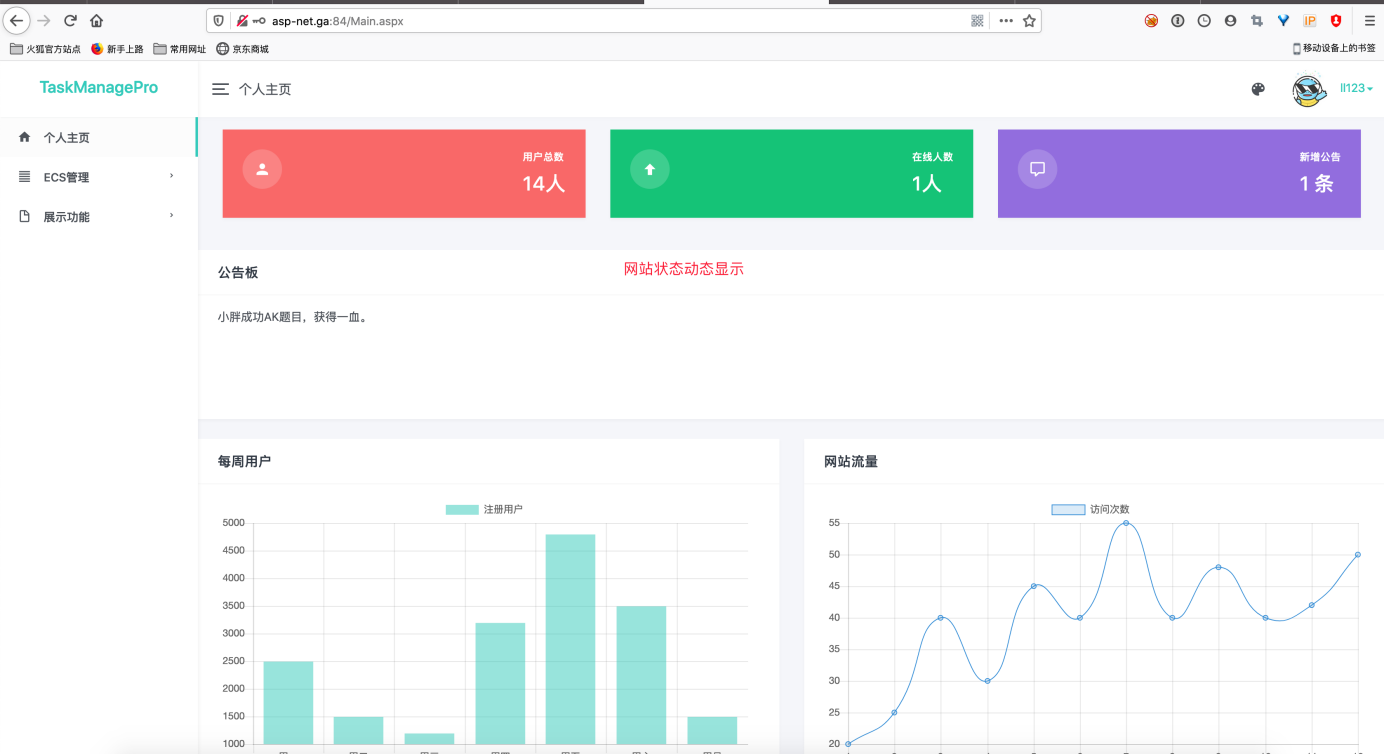


图 3.4.3 网站主界面



图 3.4.4 个人信息修改

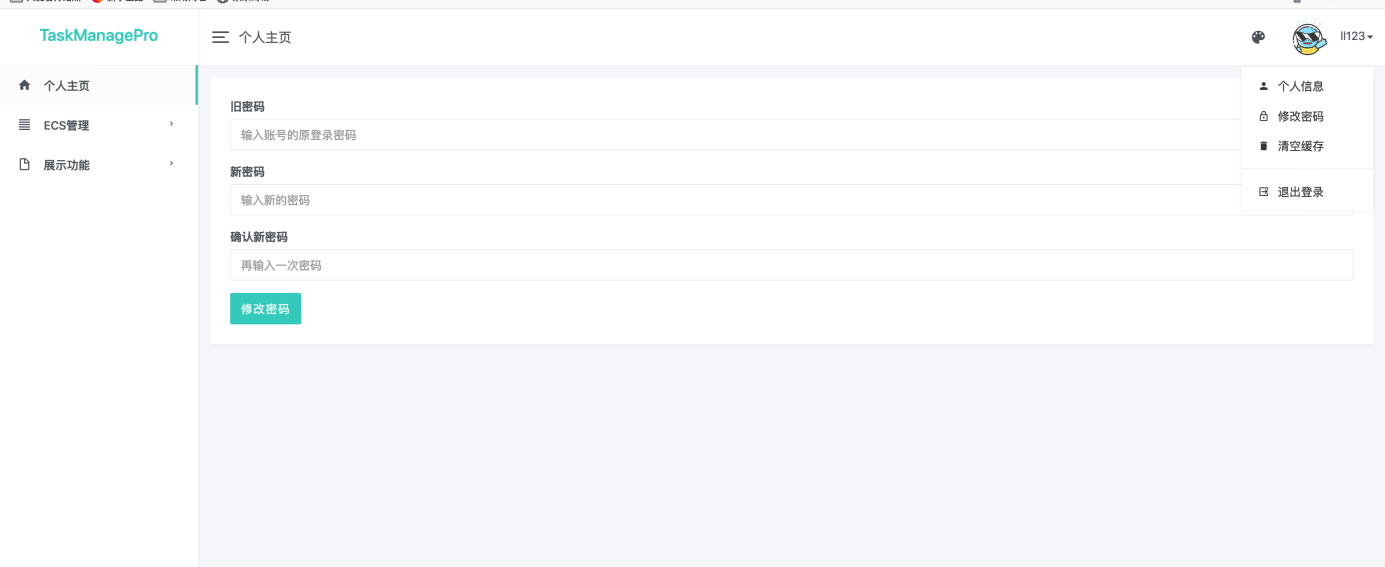


图 3.4.5 修改密码

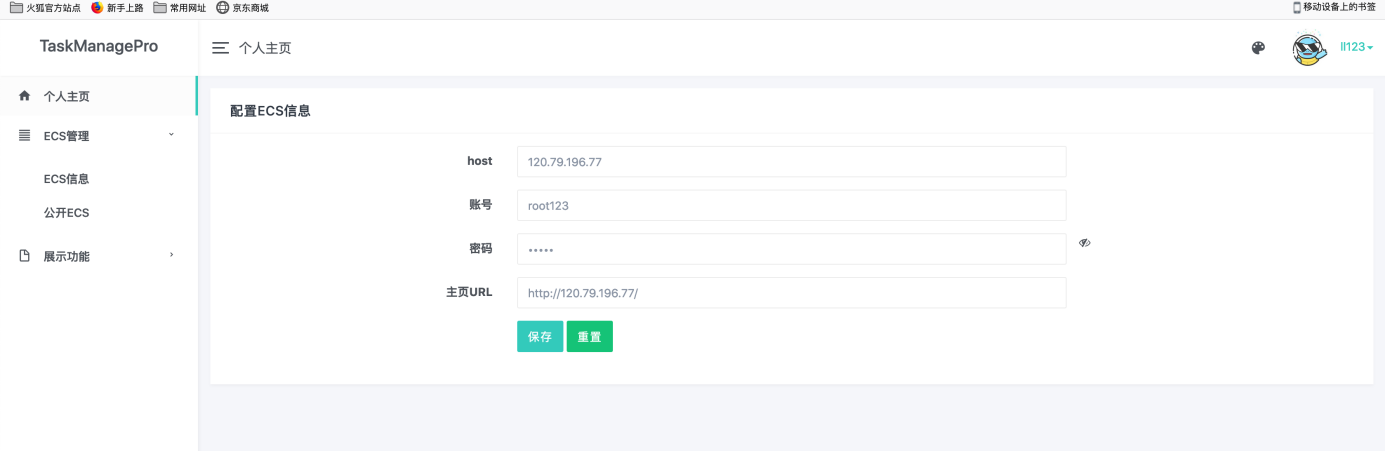


图 3.4.6 ECS信息编辑



图 3.4.7 ECS展示控制



图 3.4.8 ECS 展示列表

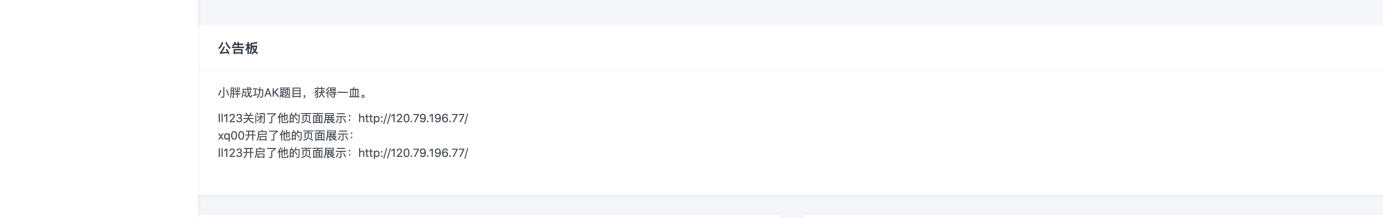


图 3.4.9 公告板信息

1. **本设计改进建议**

由于开发过程的不确定性导致实现效果与计划存在一定的差异，其二就是代码结构还是存在部分的杂糅，登录功能中没有涉及验证码来控制，目前系统支持的展示功能比较单一，后面可以尝试添加一个管理模块，用于增加展示的功能的类型，比如添加作业展示，电影资源展示等内容，并且提供对用户的增删查改功能，简化对用户的操作行为，同时在上线系统的过程中，发现系统初始化的时候，点击按钮的时候数据显示不稳定，可能是由于前端的异步请求速度与后端处理速度不匹配导致，后面可以尝试在前端异步中增加setTimeout来延时下异步的间隔，给系统足够的反应时间，从而提高网站用户的浏览体验。

1. **总结**

借助本次课程设计，我通过独立完成这个任务展示管理功能系统，不仅巩固了课程基础知识，而且加深了对ASP.NET网站运行机制的理解，也提高了自己手写代码的能力，同时通过在云端部署网站。学习了远程环境搭建和调试的技术，为今后从事开发ASP.NET实战项目积累了阶段性的经验。虽然这个系统在功能点上不是特别丰富，但是其应用的技术，涵盖了课程的核心关键点，如操作数据库的Linq技术、母版页、ajax、

空间等相关课程重点知识，可以说是麻雀虽小，但五脏俱全。虽然在设计的过程中，由于是自己设计的系统，在网上几乎没有相关的参考资料，而且网上的大量有关ASP.NET设计的资料都已经过时了，没有采用最新的Linq技术，所以笔者只能通过自己去尝试书本中的每个环节的小项目去提取出关键的代码，来编写这个系统，在这个过程中，尝试了几种前后端交互的方式，后面才决定了，采用前后端分离的思想，能够很好地兼顾前后端的特点，通过这个过程，我对如何解决开发中的问题有了实践性的认识。最后，通过这个项目，我不仅系统总结了自己所学的知识，并且提高了自己的实践能力，为今后从事ASP.NET程序开发做好了铺垫。

1. **主要参考文献**

[1] 李中华。 基于NET的模式实现与应用[D].

[2] 王国荣。 ASP网页制作教程[M].

[3] 吴晨。 ASPNET数据库项目案例导航[[J]

**附录**



前后端分离的实现代码

$('#changeProfile').on('click', function () {

lightyear.loading('show');

alert("ok");

var nickname = $("#nickname").val();

var email = $("#email").val();

//var sex = "女";

var introduce = $("#remark").val();

$.post("api/ChangProfile.aspx", { "nickname": nickname, "email": email, "introduce":introduce}, function (data) {

if (data == "ok") {

alert(data);

} else {

alert(data);

}

});

// 假设ajax提交操作

});



参考链接：

https://www.cnblogs.com/hfultrastrong/p/8798295.html

https://www.cnblogs.com/miskis/p/9816135.html