

● 高等学校21世纪计算机教材

ASP.NET

案例教程

肖金秀 何 鹏 王当文 编著



冶金工业出版社

内 容 简 介

本书主要介绍 ASP.NET 开发实例,通过这些实例系统全面地讲解 ASP.NET 的具体应用。本书的实例由初级、中级和高级三部分组成,第 1~4 章为初级应用,重点讲解如何用 ASP.NET 来开发简单的应用程序,第 5~6 章为中级应用,重点讲解如何用 ASP.NET 来开发难度较大、业务比较复杂的应用程序,第 7 章为高级应用,重点讲解了一个企业级的项目管理系统的详细开发过程,其中包括面向对象分析、设计和开发。

本书内容丰富、图文并茂、言简意赅,书中的大量实例可帮助读者快速提高其实际动手能力和开发水平。

本书既可作为高等学校相关专业和 ASP.NET 应用培训班的教材,也可作为计算机软件开发人员的参考用书。

图书在版编目 (CIP) 数据

ASP.NET 案例教程 / 肖金秀等编著. —北京: 冶金工业出版社, 2005.2
ISBN 7-5024-3696-0

I. A... II. 肖... III. 主页制作—程序设计—教材
IV. TP393.092

中国版本图书馆 CIP 数据核字 (2004) 第 143143 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 戈兰

湛江蓝星南华印务公司印刷; 冶金工业出版社发行; 各地新华书店经销

2005 年 3 月第 1 版, 2005 年 3 月第 1 次印刷

787mm × 1092mm 1/16; 21.75 印张; 504 千字; 340 页

39.00 元

冶金工业出版社发行部 电话: (010) 64044283 传真: (010) 64027893

冶金书店 地址: 北京东四西大街 46 号 (100711) 电话: (010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

前 言

一、关于 ASP.NET

ASP.NET 是一种建立在通用语言上的程序架构，是统一的 Web 开发平台，它主要为开发人员提供生成企业级 Web 应用程序所需的服务。

ASP.NET 是一个已编译的、基于 .NET 的环境，可以用任何与 .NET 兼容的语言（包括 Visual Basic.NET、C# 和 JScript.NET）开发应用程序。另外，任何 ASP.NET 应用程序都可以使用整个 .NET 框架。开发人员可以方便地获得这些技术的优点，其中包括托管的公共语言运行库环境、类型安全、继承等等。

二、本书内容结构

本书共分 7 章，各章的内容结构安排如下：

第 1 章：在线投票系统。介绍了一个简单投票系统的开发，该系统为 ASP.NET 入门实例，使读者对 ASP.NET 有一个初步的认识。

第 2 章：留言本系统。介绍了一个留言本系统的开发，该留言本比较实用，基本实现了发表留言、回复留言和留言的管理等功能。

第 3 章：新闻发布系统。介绍了一个新闻发布系统的开发，该系统主要实现了新闻类别和新闻的管理等功能。

第 4 章：论坛系统。介绍了一个功能强大的论坛的开发，该论坛可根据用户需求实现完全的自定义功能，并且介绍了大量 ASP.NET 的开发技巧。

第 5 章：在线购物系统。介绍了一个流行的在线购物系统的开发，该实例除了讲解如何开发在线购物系统，还将引导读者如何扩展思路，开发出更全面、功能更强大的在线购物系统。

第 6 章：视野车辆管理系统。介绍了一个车辆管理系统的开发，该系统可以作为办公自动化系统的一个子系统，主要实现车辆资料记录、驾驶员资料记录、用车申请、出车记录和车辆维护等功能。

第 7 章：项目管理系统。介绍了一个项目管理系统的开发，该系统是本书的重点实例，目前已经在某大型企业开始应用，并且受到用户的好评。通过该系统，读者可基本掌握如何开发一个大型应用系统，其中包括需求分析、设计、开发和系统部署等具体过程。

三、本书特点

本书实用性强、涉及面较广，如企业办公领域、在线购物和项目管理等，而且具有很大的商业价值，如在线购物系统和项目管理系统，读者可以在这些实例的基础上进行二次开发以最大限度地利用这些系统实例。

本书中的大多数实例都是按照当前较为流行的三层或多层架构体系进行设计，如三层结构包括表示层、业务层和数据层，并且部分实例还采用 UML 语言进行面向对象的分析

设计。

四、本书适用对象

本书使用面广，既可作为高等学校相关专业和 ASP.NET 应用培训班的教材，也可作为计算机软件开发人员的参考用书。

虽然经过严格的审核、精细的编辑，本书在质量上有了一定的保障，但我们的目标是力求尽善尽美，欢迎广大读者和专家对我们的工作提出宝贵建议，联系方法如下：

电子邮件：**service@cnbook.net**

网址：**www.cnbook.net**

此外，**本书涉及到的源代码可在该网站的下载中心免费下载**，还有一些其他相关书籍的介绍，可以方便读者选购参考。

编 者

2004 年 12 月

目 录

第 1 章 在线投票系统	1
1.1 系统简介	1
1.2 系统实现	1
1.2.1 投票页面	1
1.2.2 投票统计页面	2
1.2.3 投票信息存储	4
小结	5
第 2 章 留言本系统	6
2.1 系统简介	6
2.2 系统实现	6
2.2.1 数据结构	6
2.2.2 系统功能实现	6
小结	13
第 3 章 新闻发布系统	14
3.1 系统简介	14
3.2 系统实现	14
3.2.1 数据结构	14
3.2.2 系统功能实现	14
小结	24
第 4 章 论坛系统	25
4.1 系统简介	25
4.2 数据结构	25
4.3 系统功能实现	29
4.3.1 登录	29
4.3.2 问题管理	34
4.3.3 系统功能	40
4.3.4 个人信息	42
4.3.5 短信息	43
4.3.6 自定义论坛	46
4.3.7 论坛管理中心	46
4.3.8 论坛分类管理	46
4.3.9 论坛管理	49

4.3.10 用户管理	51
4.3.11 公告管理	52
4.3.12 系统短消息管理	53
小结	54
第 5 章 在线购物系统.....	55
5.1 网站模块分析	55
5.1.1 功能描述	55
5.1.2 功能结构图	57
5.1.3 技术重点	57
5.1.4 数据库结构	58
5.2 公共代码处理模块	60
5.2.1 相关代码文件	60
5.2.2 部分源代码	61
5.3 用户信息模块	80
5.3.1 相关代码文件	80
5.3.2 用户注册	80
5.3.3 用户登录和获取密码	82
5.3.4 用户信息修改	84
5.4 商品选购模块	86
5.4.1 相关代码文件	86
5.4.2 商品选购用户界面	87
5.4.3 商品选购关键技术	87
5.4.4 商品选购部分源代码	87
5.5 购物篮管理模块	91
5.5.1 相关代码文件	91
5.5.2 购物篮管理用户界面	91
5.5.3 购物篮管理部分源代码	92
5.6 地址列表管理模块	94
5.6.1 相关代码文件	94
5.6.2 地址列表管理用户界面	94
5.6.3 地址列表管理部分源代码	95
5.7 历史订单管理模块	97
5.7.1 相关代码文件	97
5.7.2 历史订单管理用户界面	98
5.7.3 历史订单管理部分源代码	98
5.8 订单提交模块	101
5.8.1 相关代码文件	101
5.8.2 订单提交用户界面	101

5.8.3 订单提交部分源代码	101
小结	109
第 6 章 视野车辆管理系统	110
6.1 需求分析	110
6.2 系统设计	110
6.2.1 系统架构	111
6.2.2 业务处理流程	111
6.2.3 功能设计	111
6.2.4 数据库设计	112
6.2.5 数据表设计	112
6.2.6 主界面设计	117
6.2.7 权限设计	118
6.3 数据存储模块与页面基类	118
6.3.1 通用数据库基类模块 DbBase	118
6.3.2 页面基类模块	126
6.4 主界面开发	131
6.4.1 页面头	132
6.4.2 访问状态	132
6.4.3 页面主体	133
6.4.4 页脚	133
6.4.5 页面样式 (CSS)	133
6.4.6 页面客户端脚本 (JS)	133
6.4.7 页面框架模板代码	133
6.5 系统管理模块	134
6.5.1 用户登录	134
6.5.2 权限管理	136
6.5.3 用户管理	139
6.5.4 修改密码	143
6.6 基础资料管理模块	144
6.6.1 部门管理	144
6.6.2 车辆管理	148
6.6.3 司机管理	152
6.7 车辆管理模块	155
6.7.1 用车申请	155
6.7.2 用车审批	157
6.7.3 出车	160
6.7.4 交车	163
小结	164

第7章 项目管理系统	165
7.1 系统概述.....	165
7.1.1 系统简介	165
7.1.2 系统特点	165
7.2 功能描述.....	166
7.2.1 项目管理	166
7.2.2 查询.....	167
7.2.3 汇总分析	167
7.2.4 数据管理	167
7.2.5 基本资料	168
7.2.6 系统管理	168
7.2.7 功能结构图	169
7.3 需求分析.....	170
7.3.1 项目及流程管理	170
7.3.2 基本信息管理	171
7.3.3 系统管理	172
7.4 系统设计.....	173
7.4.1 系统架构	173
7.4.2 类描述	175
7.4.3 交互图	177
7.4.4 数据库设计	178
7.4.5 方法描述	185
7.5 系统开发.....	188
7.5.1 配置及公共文件	188
7.5.2 系统登录	206
7.5.3 基本资料	208
7.5.4 项目管理	261
7.5.5 查询.....	320
7.5.6 汇总分析	320
7.5.7 数据管理	323
7.5.8 系统管理	330
小结	340

第 1 章 在线投票系统

本章将介绍一个简单的在线投票系统,通过此实例读者可以对 ASP.NET 的应用有一个初步的认识。

1.1 系统简介

系统可以针对某一个问题展开调查,同时提供一些选项以供用户选择,然后将投票信息保存起来,最后统计投票票数,并显示投票结果。因为本实例较为简单,所以选择用 xml 来存储投票信息,并介绍读取和更新 xml 文件的方法。

1.2 系统实现

下面介绍系统的具体实现。

1.2.1 投票页面

Vote.aspx 主要实现从 vote.xml 中读取投票选项信息,然后生成投票客户端,如图 1-1 所示。



图 1-1

程序清单: Vote.aspx.cs

本程序段的功能为:生成投票系统客户端。

```
namespace vote
{
    public class Vote : System.Web.UI.Page
    {
        private void Page_Load(object sender, System.EventArgs)
        {
            //创建 xml 对象,读取 vote.xml
            System.Xml.XmlDocument pDoc = new System.Xml.XmlDocument();
            pDoc.Load(Server.MapPath("vote.xml"));

            //生成客户端 html 页面
        }
    }
}
```

```

Response.Write("<html>");
Response.Write("<head><title>");

//读取 XML 文件的 polltitle 节点
Response.Write(pDoc.SelectSingleNode
("./candidates/polltitle").InnerText);
Response.Write("</title></head>");
Response.Write("<body>");
Response.Write("<center>");
Response.Write("<br><br><h2>");
Response.Write(pDoc.SelectSingleNode
("./candidates/polltitle").InnerText);
Response.Write("</h2>");
Response.Write("<br><br><h3>");
Response.Write(pDoc.SelectSingleNode
("./candidates/pollinstructions").InnerText);
Response.Write("</h3>");
System.Xml.XmlNodeList pCandidates;
pCandidates = pDoc.SelectNodes("./candidates/candidate");
int iCandidates;
Response.Write("<br><table border='0'>");

//读取 xml 文件, 生成选择项
for(iCandidates = 0;iCandidates<pCandidates.Count;iCandidates++)
{
    Response.Write("<tr><td>");
    string strCandidate = pCandidates[iCandidates].SelectSingleNode
("./candidateid").InnerText;
    Response.Write("<form    name=\"\"    +    strCandidate    +    \"\"
action=\"Voted.aspx?candidate=\" + strCandidate + \"\" method=\"POST\">");
    Response.Write("<a    href=\"javascript:\"    +    strCandidate    +
\".submit()\">");
    Response.Write(pCandidates[iCandidates].SelectSingleNode
("./candidatetitle").InnerText);
    Response.Write("</a>");
    Response.Write("</form>");
    Response.Write("</td></tr>");
}
Response.Write("</table>");
Response.Write("</center>");
Response.Write("</body>");
Response.Write("<html>");
Response.Write("</html>");
}
}
}

```

1.2.2 投票统计页面

Voted.aspx 主要实现将用户的投票信息写入 vote.xml, 然后再从 vote.xml 中读取投票信息并进行统计, 最后将统计结果输出显示, 如图 1-2 所示。

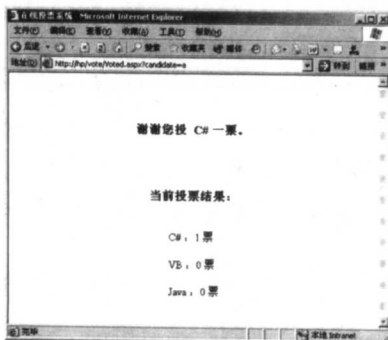


图 1-2

程序清单: Voted.aspx.cs

本程序段的功能为: 统计投票信息。

```
namespace vote
{
    public class Voted : System.Web.UI.Page
    {
        private void Page_Load(object sender, System.EventArgs)
        {
            //判断投票参数是否为空
            string strCandidateid = Request.Params["candidate"];
            if(strCandidateid == null)
            {
                Response.Write("<a href=\"vote.aspx\">请参加投票</a>");
            }
            else
            {
                //创建 xml 对象, 读取 vote.xml
                System.Xml.XmlDocument pDoc = new System.Xml.XmlDocument();
                pDoc.Load(Server.MapPath("vote.xml")); //获取用户 IP 地址
                string strVoterId = Request.UserHostAddress;
                System.Xml.XmlNodeList pAllVotes = pDoc.SelectNodes
                (".//candidates/candidate/votes/vote");
                int iAllVotes;
                bool bChad = false;

                //判断用户是否已经投票, 系统不允许重复投票
                for(iAllVotes = 0; iAllVotes < pAllVotes.Count; iAllVotes++)
                {
                    if(pAllVotes[iAllVotes].InnerText == strVoterId)
                    {
                        bChad = true;
                    }
                }
                if(!bChad)
                {
                    //存储用户投票信息
                    System.Xml.XmlNode pVoteChoice = pDoc.SelectSingleNode
                    (".//candidates/candidate[candidateid = '" + strCandidateid + "']");
                    System.Xml.XmlNode pNewVote = pDoc.CreateNode
                    (System.Xml.XmlNodeType.Element, "vote", "");
                    pNewVote.InnerText = strVoterId.Trim();
                    pVoteChoice.SelectSingleNode
                    ("votes").AppendChild(pNewVote);
                    pDoc.Save(Server.MapPath("vote.xml"));
                    Response.Write("<html>");
                    Response.Write("<head><title>");
                    Response.Write(pDoc.SelectSingleNode
                    (".//candidates/polltitle").InnerText);
                    Response.Write("</title></head>");
                    Response.Write("<body>");
                    Response.Write("<center>");
                    Response.Write("<br><br><table cellpadding=20>");
                    Response.Write("<tr>");
                    Response.Write("<td><h3>");
                    Response.Write(" 感谢您投 " + pVoteChoice.SelectSingleNode
                    ("candidateid").InnerText + " 一票。");
                    Response.Write("</h3></td>");
                    Response.Write("</tr>");
                    Response.Write("</table>");
                    Response.Write("<br><br><h3>当前投票结果: </h3>");
                    System.Xml.XmlNodeList pCandidates;
                    pCandidates = pDoc.SelectNodes(".//candidates/candidate");
                    int iCandidates;
```



```
</candidates>
```

小结

本章简单介绍了在线投票系统，通过实例初步讲解了 ASP.NET 的具体应用，为下面进一步学习和掌握 ASP.NET 打下基础。

第 2 章 留言本系统

本章将介绍一个留言本系统，通过此实例可以进一步了解 ASP.NET，并对其在数据库方面的应用有一个初步的认识。

2.1 系统简介

此系统主要实现了留言本的基本功能，如发表留言、回复留言和删除留言。系统的后台数据库为 Access，本章还将介绍数据库的一些基本操作，如查询、插入和删除。

2.2 系统实现

下面将具体介绍系统的实现，主要包括数据结构和系统功能实现。

2.2.1 数据结构

系统数据结构的详细信息如下：

表名：Content（留言内容），如表 2-1 所示。

表 2-1 留言内容

字段名称	数据类型	主键	外键	非空	默认值	字段说明
ID	自动编号	True		True		留言编号
UserName	文本			True		用户名
DTime	日期/时间			True		时间
Cont	备注			True		内容
EEmail	文本					邮件地址
Reply	备注					回复内容

2.2.2 系统功能实现

系统主要功能为：查看留言、发表留言、回复留言和删除留言。

1. 查看留言

Default.aspx 是留言本首页，如图 2-1 所示。

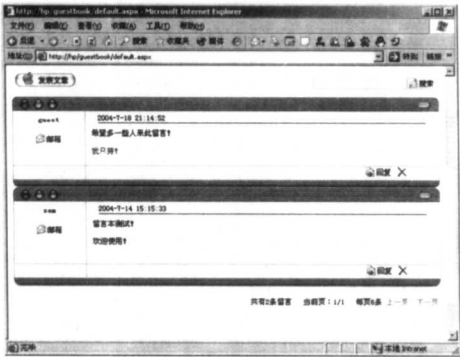


图 2-1

主要实现从数据库中读取留言信息，然后将留言进行输出显示。

对于用户留言，系统可分页进行显示，而且每页显示的留言条数可以在 Web.config 中进行自定义，单击“上一页”和“下一页”按钮可在不同页面之间进行切换。

在首页，单击“发表文章”按钮可进入发表留言页面进行留言；还可单击“回复”按钮回复留言，或者单击×按钮进行删除留言；如果留言过多，还可输入关键字来查询所需要的留言。

程序清单：Web.config

本程序段的功能为：自定义留言本属性。

```
<appSettings>
<!--设置留言本名称-->
<add key="GuestBookName" value=".NET 留言本" />
<!--设置数据库路径-->
<add key="DBPath" value="db\guestbook.mdb" />
<!--设置每页显示留言条数-->
<add key="PageNum" value="6" />
<!--设置管理员密码-->
<add key="PassWord" value="Admin" />
</appSettings>
```

程序清单：Default.aspx

本程序段的功能为：显示留言信息。

```
//每页显示留言条数
int pagesize= System.Convert.ToInt32
(ConfigurationSettings.AppSettings["PageNum"]);
int nowpage;

void Page_Load(object obj,EventArgs)
{
    //页面初始化
    if(!IsPostBack)
    {
        dodatabind("select * from content order by ID desc","select count(*) from
content",1);
        pos2.Visible=false;
    }
}

//根据关键字查询留言
void dosearch(object obj,ImageClickEventArgs)
{
    if(keywordTB.Text!="")
    {
        searchlabel.Text=keywordTB.Text;
        searchlabel2.Text=keywordTB.Text.Replace("
","&nbsp;").Replace("<","&#34;").Replace("'", "&#39;").Replace("<","&#60;").Replace(">","&#62;");
        dodatabind("select * from content where cont like @keyword order by ID
desc","select count(*) from content where cont like @keyword",1);
        pos2.Visible=true;
    }
    else
    {
        Response.Redirect("default.aspx");
    }
}
```

```

    }

    //向上一页
    void goprepage(object obj,EventArgs)
    {
        dodatabind("select * from content where cont like @keyword order by ID
        desc","select count(*) from content where cont like @keyword",
            System.Convert.ToInt32(nowpagelabel.Text)-1);
    }

    //向下一页
    void gonextpage(object obj,EventArgs)
    {
        dodatabind("select * from content where cont like @keyword order by ID
        desc","select count(*) from content where cont like @keyword",
            System.Convert.ToInt32(nowpagelabel.Text)+1);
    }

    //从数据库获取留言信息
    void dodatabind(string sql,string countsq, int pagenum)
    {
        string connstr="Provider=Microsoft.Jet.Oledb.4.0;Data
        Server.MapPath(ConfigurationSettings.AppSettings["DBPath"])+";";
        Source="+

        OleDbConnection conn=new OleDbConnection(connstr);
        conn.Open();

        //计算留言总数
        int recnum;
        OleDbCommand cmd = new OleDbCommand(countsq,conn);
        OleDbParameter parm=cmd.Parameters.Add("@keyword",OleDbType.WChar);
        parm.Value="%" + searchlabel.Text + "%";
        parm.SourceVersion = DataRowVersion.Original;
        OleDbDataReader dr = cmd.ExecuteReader();
        if(dr.Read())
            recnum=dr.GetInt32(0);
        else
            recnum=0;
        dr.Close();

        //计算留言总页数
        int allpage=((recnum%pagesize)==0)?(recnum/pagesize):((recnum/pagesize)+1);

        //创建数据库连接,读取留言信息
        DataSet ds=new DataSet();
        cmd=new OleDbCommand(sql,conn);
        parm=cmd.Parameters.Add("@keyword",OleDbType.WChar);
        parm.Value="%" + searchlabel.Text + "%";
        parm.SourceVersion = DataRowVersion.Original;
        OleDbDataAdapter da=new OleDbDataAdapter();
        da.SelectCommand=cmd;
        int startnum=(pagenum-1)*pagesize;
        da.Fill(ds,startnum,pagesize,"content");
        rl.DataSource=ds.Tables["content"].DefaultView;
        rl.DataBind();

        //显示状态
        pagesizelabel.Text=pagesize.ToString();
        countnum.Text=recnum.ToString();
        nowpagelabel.Text=pagenum.ToString();
        allpagenumbabel.Text=allpage.ToString();
        nextpage.Enabled=!(Boolean)(pagenum>=allpage);
        prepage.Enabled=!(Boolean)(pagenum<=1);

        //关闭连接
        conn.Close();
        conn.Dispose();
        cmd.Dispose();
    }

```

```

        da.Dispose();
        ds.Dispose();
    }

```

2. 发表留言

Add.aspx 的主要功能为发表留言，将用户发表的留言信息写入数据库，如图 2-2 所示。

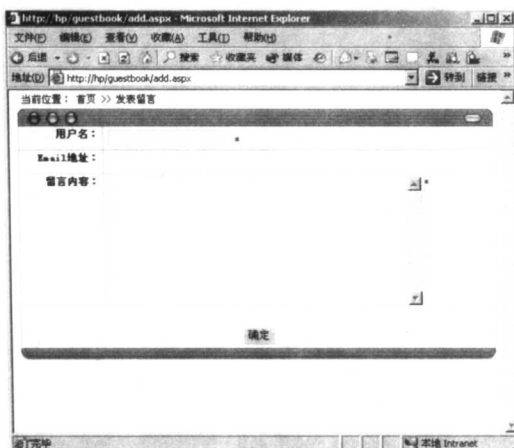


图 2-2

发表留言时，用户名和留言内容不允许为空。留言填写完成之后，单击“确定”按钮即可将留言保存。此时还可以单击页面顶部的导航栏以返回首页。

程序清单：Add.aspx

```

*****
本程序段的功能为：发表留言。
*****
void additem(Object Sender,EventArgs)
{
    if (Page.IsValid==true)
    {
        //创建数据库连接
        string dbpath=Server.MapPath (ConfigurationSettings.AppSettings
["DBPath"]);
        string connstr="Provider=Microsoft.Jet.Oledb.4.0;Data Source="+ dbpath;
        OleDbConnection conn=new OleDbConnection(connstr);
        conn.Open();

        //更新数据库，写入留言
        string sql = String.Format("insert into
content(username,DTime,content,email)
values(@username,'{1}',@content,'{3}')","",DateTime.Now,"",email.Text);
        OleDbCommand updatecomm=new OleDbCommand(sql,conn);
        updatecomm.Parameters.Add("@username",
OleDbType.VarChar,20).Value=username.Text;
        updatecomm.Parameters.Add("@content",
OleDbType.WChar).Value=content.Text;
        updatecomm.ExecuteNonQuery();

        //关闭数据库连接
        conn.Close();
        conn.Dispose();
        Response.Redirect("default.aspx");
    }
}

```