

第 12 章 用户注册登录系统实例

这一章将以一个完整的 JSP 实例来演示以上章节所介绍的内容。其实，在本书的第 3 章已经创建了第一个 JSP 应用，用来说明 JSP 运行原理和执行机制。本章的实例还会向读者介绍如何使用 Eclipse+Lomboz 工具快捷、方便地架构其一个 JSP 应用。

学完本章内容之后，读者应该了解 Eclipse+Lomboz 在 Web 开发中的使用，包括 Web 模块的创建、通过 Lomboz 如何方便地启动和停止 Web 服务器，以及发布 Web 应用。

本章要点包括以下内容：

- ☐ 如何使用 Lomboz 启动 Web 服务器
- ☐ 如何使用 MySQL 客户端管理工具来创建数据库表
- ☐ 详细介绍一个完成的 JSP 实例
- ☐ 了解 JSP 项目的完整开发过程

12.1 实例介绍

本章选择用户注册登录系统作为实例，是因为它是一个典型的问题。该系统会涉及到表单提交和处理、数据库操作以及使用 session 存储用户登录信息这些基本功能。

首先针对非会员用户来说，需要进行注册。这时，JSP 程序需要将用户提交的注册信息进行处理，即进行数据库存储操作。

当用户进行登录时，后台 JSP 程序会进行身份验证，这需从数据库读取出该用户信息。首先，通过填写的用户名在数据库中查找该用户信息，如果不存在，即表示该填写的用户不存在；如果从数据库中读取的该用户密码和登录页面所填写的密码不符合，即表示登录失败（显示密码不正确）。

用户登录信息还需要使用 session 内置对象（有关 session 的概念在前面已经有所介绍）存储起来，以便用户可以在该 Web 站点中的所有页面上进行相应权限的操作。登录用户可以浏览器首页的产品信息，非登录用户则会跳转到登录页面进行登录或者注册。

12.2 用户注册登录系统设计

首先需要对该实例进行相应的功能和数据库设计，一般通过图表的形式将系统功能描述清楚。该系统主要包括三个部分：首页浏览、登录和注册。其中数据库表包括用户信息表和产品信息表。

12.2.1 系统功能图

该系统需要创建的页面有如下：

- ☐ index.jsp：首页产品浏览，非登录用户访问会跳转到 login.jsp 登录页面；
- ☐ login.jsp：用户登录页面，其中包括注册链接；

- ❑ chek_login.jsp: 进行相应用户身份验证;
- ❑ logout.jsp: 用户注销操作;
- ❑ register.jsp: 用户注册页面;
- ❑ do_register.jsp: 将页面注册信息存入数据库。

下面通过图直观地表示这些页面之间的关系，如图 12.1 所示。

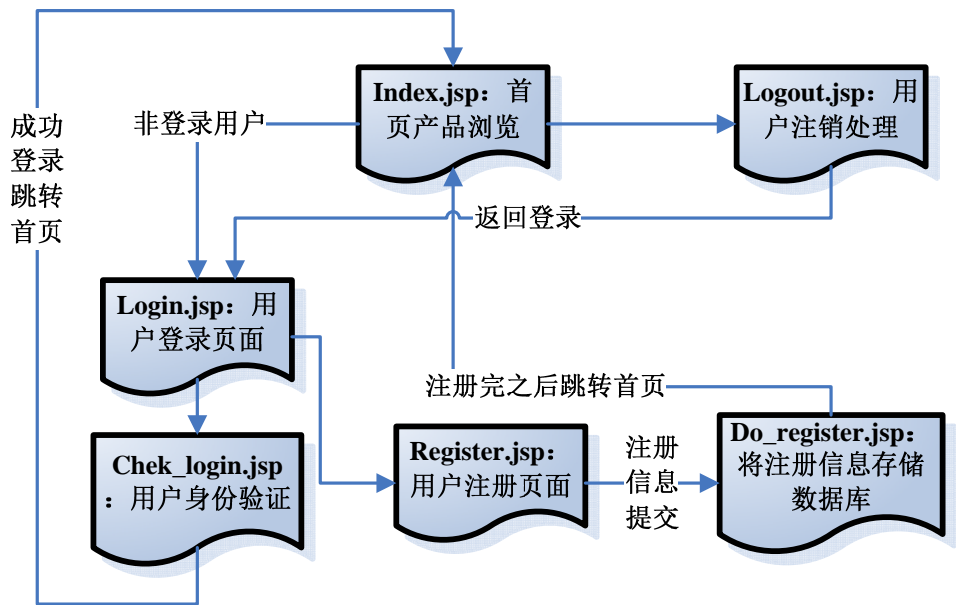


图 12.1 系统功能图

从图中可以看出，用户访问站点的 index.jsp 页面，首先会判断用户是否登录，如果处于登录状态，则进行产品浏览；如果未登录，会跳转登录页面。用户成功登录之后，会返回到首页。在用户登录页面，如果用户还没有账户，可以单击“注册”链接进行用户注册。同样，注册成功之后，会返回到首页并默认设置用户为登录状态。在首页，用户也可以单击“注销”来安全退出登录状态。

12.2.1 数据库表设计

由于本系统比较简单，所建立的表也比较简单，如下：

- ❑ users: 该数据库表存储用户基本信息，其中包括用户名、用户密码、真实姓名、性别、出生年月以及个人描述等字段。
 - ❑ products: 产品信息表，其中包括产品编号、产品名称、产品价格以及产品描述组成。
- 有关 users 表更详细字段介绍如表 12.1 所示。

表 12.1 users 表

字段	数据类型	名称	描述
USER_ID	varchar(50) *	用户名	用户名惟一，最长50字节
PASSWORD	Varchar(20)	密码	密码最长20个字节
NAME	Varchar(8)	用户真实姓名	最长8个字节
SEX	char	用户性别	Char字符型
BIRTH	Long	出生年月	Long类型存储日期
DESCRIPTION	Varchar(256)	个人描述	最多128个汉字，即256字节

products 表的各字段描述如表 12.2 所示。

表 12.1 products表

字段	数据类型	名称	描述
PRODUCT_ID	bigint(20)*	商品编号	惟一，最长50字节
PRODUCT_NAME	Varchar(50)	商品名称	商品名称
PRICE	Float	商品价格	商品价格，浮点类型
DESCRIPTION	Varchar(256)	商品描述	最多128个汉字，即256字节

12.3 创建实例的一个J2EE项目

该实例创建的 Web 模块取名为 Register。其中使用的数据库为 MySQL，Web 服务器为 Tomcat。按照第九章使用 Eclipse+Lomboz 构建 J2EE 项目的方法，来创建一个名为 MyRegister 的项目。

12.3.1 使用Lomboz启动Web服务器

一个 JSP 应用在运行之前，首先需要将它发布的 Web 服务器上，并成功启动起 Web 服务器。有关 Tomcat 服务器的启动和停止方法已经在第 2 章的 2.2.3 小节作了详细介绍。这一小节主要介绍读者如何使用 Lomboz 插件更加便捷地启动和停止 Web 服务器。

在如图 12.2 所示的 MyRegister 项目导航栏中，右击 Register 模块，弹出如图 12.2 所示的下拉菜单选项。选择“Lomboz J2EE”|“Run Server”命令，启动 Tomcat 服务器。在控制台会显示 Tomcat 启动的信息，如果成功启动会有正确的提示信息，如图 12.3 所示。

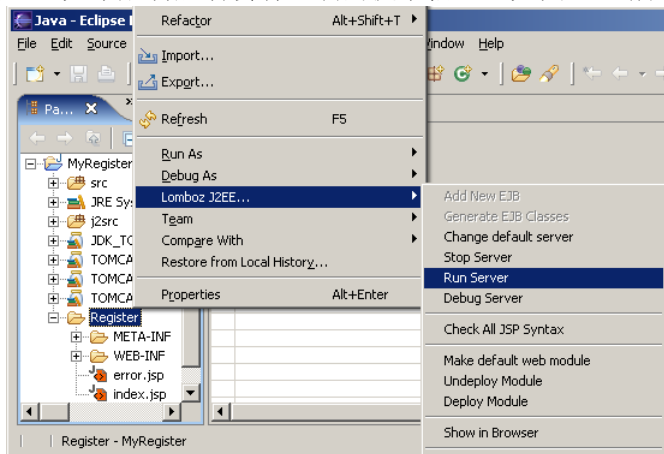


图 12.2 右键菜单选项

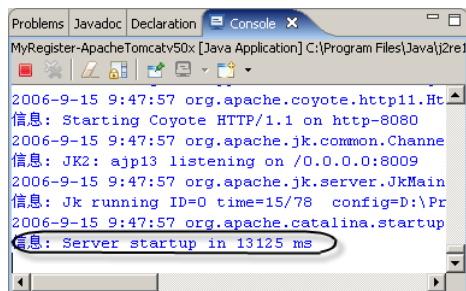


图 12.3 Tomcat 服务器成功启动

如图 12.2 所示的右键菜单中还有其他一些常用的命令。

- ❑ “Stop Server”选项：停止 Tomcat 服务器，控制台会显示成功停止 Tomcat 的信息。
- ❑ “Change default server”选项：修改默认 Web 模块的服务器，即可以随时选择运行该 Web 应用的服务器，例如可以选择 Tomcat，也可以选择 Weblogic（如果已经在 Lomboz 中存在并设置）。
- ❑ “Debug Server”选项：使用调试的方式来启动 Tomcat 服务器，在 Web 开发中更常用。该指令相当于停止和重启服务器两步操作，因为有些 Web 修改需要重新启动服务器才能有效。
- ❑ “Deploy Module”选项：把此处开发的 Web 模块发布到 Tomcat 服务器上，具体方法会在后面

介绍。有了该工具就不需要再像第 3 章所示的手动来发布 Web 模块。

- ❑ “Undeploy Module” 选项：删除已经发布到 Tomcat 服务器上的 Web 模块，即卸载掉该 Web 模块。

12.4 数据库表创建

经过以上系统功能和数据库设计之后，这一小节进入开发阶段，包括数据库表的生成和页面代码的编写。在实际项目开发过程中，一般还会使用到 CVS（版本控制软件），这部分知识留在实战的篇幅进行介绍。

12.4.1 创建数据库mydb

本实例需要创建 users 表和 products 表。首先使用 MySQL Query Browser 客户端管理工具创建一个名为 mydb 的数据库。创建步骤如下：

（1）在“Schemata”栏中点击右键，在下拉菜单中选择“Create New Schema”命令，如图 12.4 所示。弹出如图 12.5 所示的对话框。

（2）在输入框中填入所需创建的数据库名称 mydb。然后单击“ok”按钮完成创建。

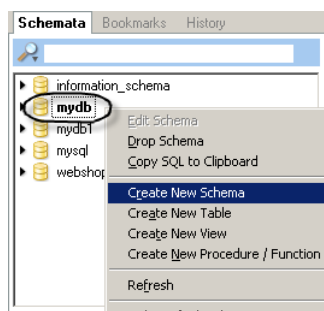


图 12.4 创建数据库

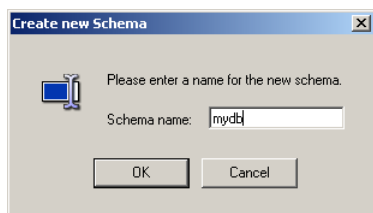


图 12.5 输入数据库名

12.4.2 创建users表

创建数据库表有两种方法：一种通过编写 SQL 语句手动创建，第二通过 MySQL Query Browser 的可视化界面自动创建。

1. 执行 SQL 语句创建

在 MySQL Query Browser 客户端管理工具的 SQL 执行框中输入 SQL 语句来生成 users 表：

```
DROP TABLE IF EXISTS `mydb`.`users`;  
CREATE TABLE `mydb`.`users` (  
  `USER_ID` varchar(50) NOT NULL default "",  
  `PASSWORD` varchar(20) NOT NULL default "",  
  `NAME` varchar(8) NOT NULL default "",  
  `SEX` char(1) NOT NULL default '男',  
  `BIRTH` bigint NOT NULL,  
  `DESCRIPTION` varchar(256) NOT NULL default "",  
  PRIMARY KEY (`USER_ID`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

由于 SQL 语句的编写规则不是本书讲解的重点，所以有关数据库创建等详细语法请读者查阅相关书籍。

2. 可视化创建数据库表

另外，可以在图 12.4 中，右击 mydb 数据库，选择“create new table”命令，出现如图 12.6 所示的界面。

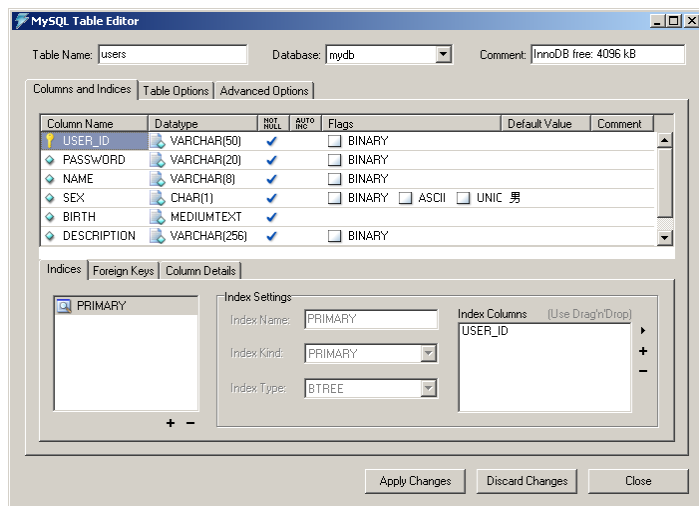


图 12.6 创建数据库表 users

其中还需要设置数据库表的字符编码。在图 12.6 所示的界面中，选中“Table Options”选择卡，出现如图 12.7 所示的界面，在“CharSet”选择框中选择 utf8 选项。

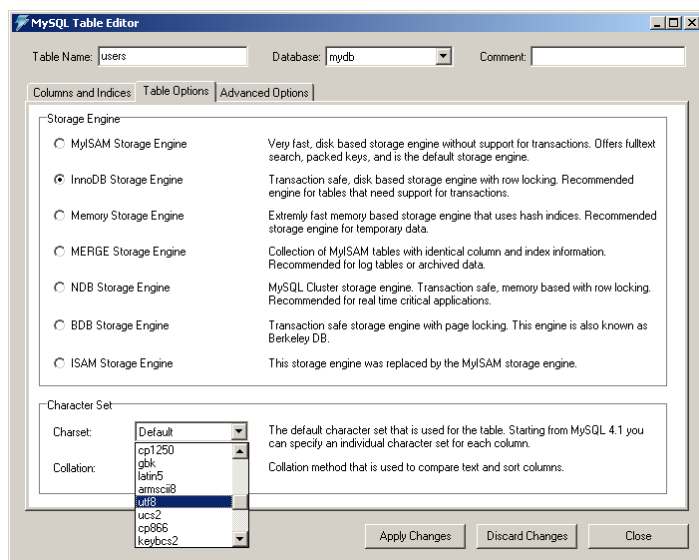


图 12.7 字符编码的选择

然后单击“Apply Changes”按钮完成数据库表 users 的创建。

12.4.3 创建products表

同样可以使用以上介绍的两种方法来创建该数据库表。

1. 执行 SQL 语句创建

创建数据库表 products 的 SQL 语句如下：

```
DROP TABLE IF EXISTS `mydb`.`products`;
CREATE TABLE `mydb`.`products` (
  `PRODUCT_ID` bigint(20) unsigned NOT NULL auto_increment,
  `PRODUCT_NAME` varchar(50) NOT NULL default "",
  `PRICE` float NOT NULL default '0',
  `DESCRIPTION` varchar(256) NOT NULL default "",
  PRIMARY KEY (`PRODUCT_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

然后单击“Execute”命令执行以上 SQL 语句。

2. 可视化创建数据库表

同样可以选择“create new table”命令来创建该数据库表。创建的界面如图 12.8 所示。

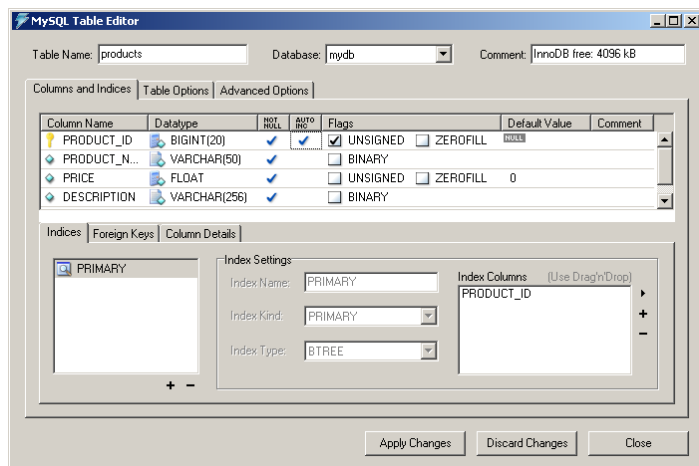


图 12.8 创建数据库表 products

单击“Apply Changes”按钮完成数据库表 products 的创建。

12.4.4 配置MySQL数据库驱动包

上一章已经对连接数据库的驱动器知识作了详细介绍。在对数据库进行操作之前，还需要安装连接数据库的驱动包，使用到的包即为第十章 10.2.3 小节下载的 mysql-connector-java-3.1.12.zip 包文件。将该压缩文件解压，得到 mysql-connector-java-3.1.12-bin.jar 文件，然后将该文件复制到 Register 模块的 WEB-INF\lib 目录下。Web 服务器会默认调用 WEB-INF\lib 目录下的 Jar 包文件（其实是由一系列 class 打包而成的 jar 文件）。然后在 Eclipse 中进行配置，即将驱动包文件加入到 Eclipse 库中，这样 Eclipse 才能识别，但是没有这一步，Web 服务器是能识别到该驱动包。具体配置过程如下：

(1) 右击 MyRegister 项目，选择“Properties”命令，弹出如图 12.9 所示的对话框。

(2) 在左侧列表框中选择“Java Build Path”选项，然后选择右侧的“Library”选择卡。再单击“Add Jars”按钮将放置在 Register 模块 WEB-INF\lib 目录下的 mysql-connector-java-3.1.12-bin.jar 包文件加入

到 Eclipse 库中。

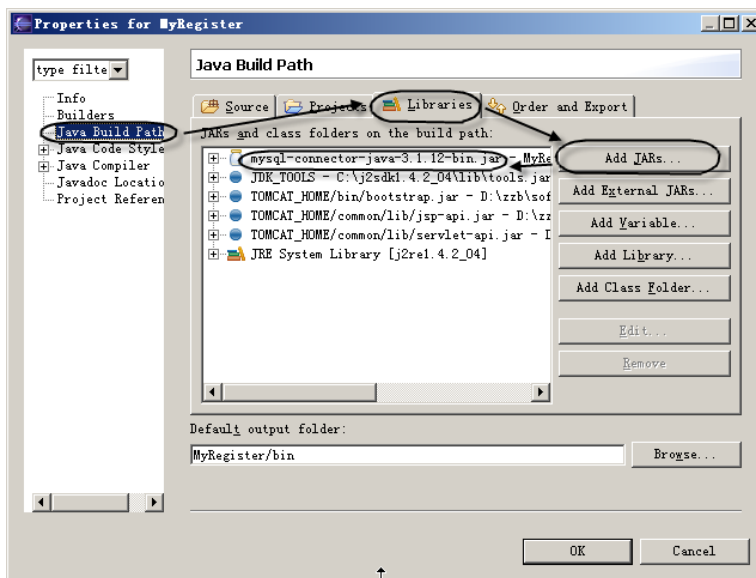


图 12.9 添加数据库驱动包文件

(3) 单击“OK”按钮完成添加。

配置完 JDBC 驱动包之后，就可以使用显示的 `Class.forName()` 来装载该驱动器了。然后建立数据库连接，再进行相应的数据库操作。

12.5 实例开发

以上完成了 Web 项目的创建和环境的配置，以及数据库表的建立。下面编写代码实现各页面功能。通过该实例的介绍来加深读者对 JSP 的运行机制、语法以及开发过程的理解。

12.5.1 创建首页 index.jsp

首页展示产品信息，并提供“登录”和“注销”链接。详细的实现代码如下：

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*" errorPage="" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>注册登录系统首页</title>
<link rel="stylesheet" href="register.css" type="text/css"/>
</head>
<%
String UserId = (String)session.getAttribute("user");           //从 session 中获取用户 ID
if(UserId == null || UserId == "")                               //判断用户 ID 是否为空，即判断是否登录
    response.sendRedirect("login.jsp");                           //未登录则跳转到 login.jsp 登录页面

String product_name = "";                                         //定义产品名称变量
```

```

float price;                                //定义产品价格变量
String description = "";                     //定义产品描述变量

Connection con = null;                       //定义数据库连接变量
Statement sm = null;                         //定义处理 SQL 语句的 Statement 变量
ResultSet rs = null;                         //定义结果集变量
try{
    Class.forName("com.mysql.jdbc.Driver").newInstance(); //装载 JDBC 驱动
    String url = "jdbc:mysql://localhost/mydb";           //定义数据库连接 URL
    con = DriverManager.getConnection(url,"root","12345"); //获取数据库连接实例
    sm = con.createStatement();                           //获取 Statement 的实例
    rs = sm.executeQuery("select * from products");        //执行查询语句，并返回给结果集
}catch(Exception e){
    e.printStackTrace();
}
%>
<body>
<table width="80%" align="center">
    <tr>
        <td class="title">&nbsp;&nbsp;&nbsp;用户注册登录系统==<span class="title1">产品展示</span>==</td>
        <td style="width:150; height:20; vertical-align:middle; text-align:center;"><%=UserId%>&nbsp;&nbsp;&nbsp;您
        好!&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href="login.jsp">登录</a>&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;<a href="logout.jsp">注销</a></td>
    </tr>
</table>
<hr align="center" width="75%" color="#990000" size="1"/>
<table width="80%" align="center">
    <tr><td height="5" colspan="2"></td></tr>
    <%
        while(rs.next()){                                //循环输出产品信息
            product_name = rs.getString("PRODUCT_NAME");
            price = rs.getFloat("PRICE");
            description = rs.getString("DESCRIPTION");
        %>
    <tr>
        <td>&nbsp;&nbsp;&nbsp;产品名称: <%=product_name%></td><td width="40%">&nbsp;&nbsp;&nbsp;报价: <span style="color:#FF0000;
font-style:italic;"><%=price%></span></td>
    </tr>
    <tr>
        <td colspan="2">&nbsp;&nbsp;&nbsp;描述: <%=description%></td>
    </tr>
    <tr><td height="5" colspan="2"><hr size="3" width="10%"/></td></tr>
<%
    }
Finally                                     //最后一定执行的语句
{
    if(rs!=null){
        try{ rs.close();}catch(Exception e){e.printStackTrace();}
    }
    if(sm!=null){
        try{ sm.close();}catch(Exception e){e.printStackTrace();}
    }
}

```



```
    }  
    if(con!=null){  
        try{ con.close();}catch(Exception e){e.printStackTrace();}  
    }  
}  
%>  
</table>  
<div align="center" class="tail">2006==本公司版权所有</div>  
</body>  
</html>
```

代码说明：代码第一行设置了 Page 指令中的多个属性，这在第七章中都详细介绍过。第二行是 W3 规定的文档类型定义（DTD）。

通过该 JSP 代码，读者应该更加清晰认识到 JSP 即为 HTML 标签和 JSP 脚本元素（在第七章主要介绍了三种脚本元素，例如本代码中的 `<%=Expression%>` 表达式格式、嵌入在 `<%` 和 `%>` 之间的 Java 程序片断）组成。

程序首先调用 session 内置对象的 `getAttribute()` 方法来获取到登录用户的 ID，如果返回为 NULL，则表示该用户处于未登录状态。未登录页面跳转到登录 `login.jsp` 页面。

读者如果已经认真阅读了本书的第 11 章，应该对程序中的数据库操作代码不陌生。首先使用 `Class.forName()` 显示方式来加载数据库的 JDBC 驱动器（如果没有设置前面提到的 `mysql-connector-java-3.1.12-bin.jar` 驱动包文件，这段代码会报错）。已经知道 JDK 是已经兼容 JDBC-ODBC 桥驱动，这里也可以使用桥驱动，这时只要设置 ODBC 的数据源（在十一章中已经介绍配置方法）。

驱动加载完之后，即可以使用 `DriverManager` 管理器中的 `getConnection()` 方法来获取到一个数据库连接实例。然后结合 `Statement` 和 `ResultSet` 来获取到 SQL 执行结果。

在使用 `While` 语句循环输出产品信息的代码中，读者应该可以看到 Java 程序片断能和 HTML 格式代码很好的结合起来使用，而不像 `Servlet` 需要太多的 `System.out.println()` 语句来输出结果。

最后执行 `Finally{}` 中代码，来释放连接数据库的各类资源。这一步是必须要做，因为计算机资源是有限的，那就需要程序在使用完资源之后及时归还给系统。

该页面运行的效果如图 12.10 所示。

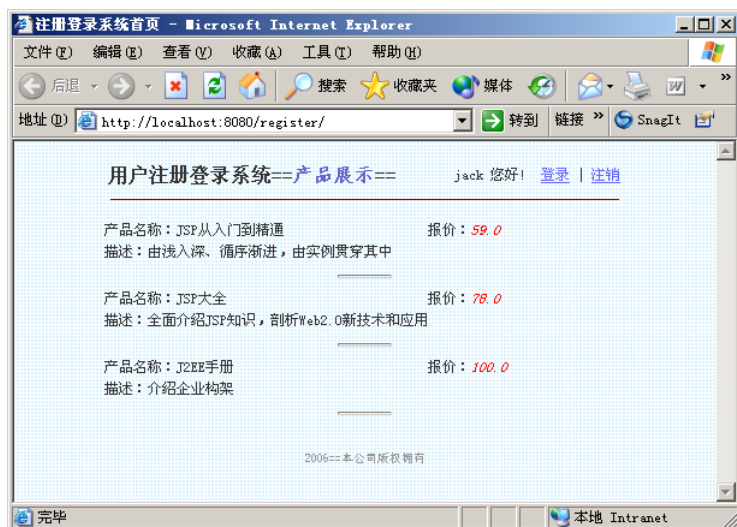


图 12.10 首页运行效果

该页面运行效果是用户成功登录之后的界面。如果未登录，页面会直接跳转到登录页面，这个页面是看不到的。下面创建用户注册页面 register.jsp。

12.5.2 创建用户注册页面register.jsp

用户需要在注册页面上填写用户名、密码（包括重试密码）、姓名、性别、出生年月以及个人简单描述等信息，然后提交后台处理。该页面的详细代码如下：

```
<%@ page contentType="text/html; charset=gb2312" language="java" errorPage="" %>
<HTML>
<HEAD><TITLE> 商行一注册 </TITLE></HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=gb2312">
<link rel="stylesheet" href="register.css" type="text/css"/>
<SCRIPT LANGUAGE=javascript>
<!--
function checkform(){
if(document.form.ID.value=="||(document.form.ID.value).indexOf("/")||(document.form.ID.value).indexOf("."){
    alert("用户名为空或者有非法字符！");
    document.form.ID.focus();
}
else if(document.form.password.value=="||(document.form.password.value).length<6){
    alert("密码为空或者太短，为了你的安全请填写！");
    document.form.password.focus();
}
else
f(document.form.rpassword.value=="||(document.form.rpassword.value)!=(document.form.password.value)){
    alert("重置密码为空，或者和密码不一致！");
    document.form.rpassword.focus();
}
else if(document.form.name.value==""){
    alert("您的姓名为空！");
    document.form.name.focus();
}
else{
    document.form.submit();
}
}
}
//-->
</SCRIPT>
<BODY>
<table width="80%" border="0" cellspacing="0" cellpadding="0" align="center">
<tr><td height="5">&nbsp;</td></tr>
<tr>
<td>
<div align="center"><font size=3><Strong>用户注册</strong></font><br>
<hr width="80%" size="1"></div>
</td>
</tr>
<tr>
<td>
```

```

<form method=post action="do_register.jsp" name="form">
<table width="80%" border="0" cellspacing="0" cellpadding="0" align="center">
  <tr height="30">
    <td width="20%" align="right">&nbsp;用户名: </td>
    <td width="40%" align="left">&nbsp;<input type="text" name="ID" size="15">&nbsp;</td>
    <td width="40%" align="left">&nbsp;<font color="#0099FF">请不要包含'/'、'\'、','等字符</font></td>
  </tr>
  <tr height="30">
    <td width="20%" align="right">&nbsp;密码: </td>
    <td width="40%" align="left">&nbsp;<input type="Password" name="password" size="15">&nbsp;</td>
    <td width="40%" align="left">&nbsp;<font color="#0099FF">密码要足够长，以确保安全</font></td>
  </tr>
  <tr height="30">
    <td width="20%" align="right">&nbsp;重新输入一次密码: </td>
    <td width="40%" align="left">&nbsp;
      <input type="Password" name="rpassword" size="15">&nbsp;
    </td>
    <td width="40%" align="left">&nbsp;</td>
  </tr>
  <tr height="30">
    <td width="20%" align="right">&nbsp;姓名: </td>
    <td width="40%" align="left">&nbsp;<input type="text" name="name" size="15"></td>
    <td width="40%" align="left">&nbsp;</td>
  </tr>
  <tr height="30">
    <td width="20%" align="right">&nbsp;性别: </td>
    <td width="40%" align="left">&nbsp;
      <input type="radio" name="sex" value="男" checked>男
      <input type="radio" name="sex" value="女">女
    </td>
    <td width="40%" align="left">&nbsp;</td>
  </tr>
  <tr height="30">
    <td width="20%" align="right">&nbsp;出生年月: </td>
    <td width="40%" align="left">&nbsp;
      <select name="year"><% for(i=1970;i<=2000;i++) {%>
        <option value=<%=i%>><%=i%></option><%=i%></select>年
      <select name="mouth"><% for(i=1;i<=12;i++) {%>
        <option value=<%=i%>><%=i%></option><%=i%></select>月
      <select name="day"><% for(i=1;i<=31;i++) {%>
        <option value=<%=i%>><%=i%></option><%=i%></select>日
      </td>
    <td width="40%" align="left">&nbsp;</td>
  </tr>
  <tr height="30">
    <td width="20%" align="right">&nbsp;个人说明: </td>
    <td width="40%" align="left">&nbsp;
      <textarea name="description" cols="30" rows="5"></textarea></td>
    <td width="40%" align="left">&nbsp;</td>
  </tr>

```

[illegible]

代码说明:该页面使用表单(form)来提交用户所填写的个人信息。但是在提交之前,会执行 JavaScript 的 `checkform()` 方法来对各表单内容进行简单校验(例如是否为空、字符长度是否符合规定等)。该页面运行的效果如图 12.11 所示。

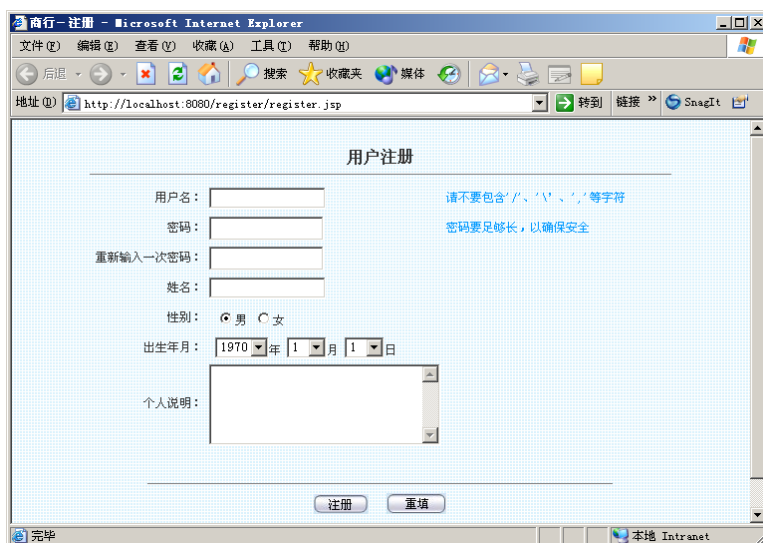


图 12.11 用户注册页面效果

12.5.3 创建用户注册处理页面do_register.jsp

该页面处理用户注册的信息，即将用户填写的个人信息存储到数据库表 `users` 中。如果注册成功，将用户 ID 存储在 `session` 中，页面并跳转到首页 `index.jsp`；注册失败，则重新跳转到 `register.jsp` 页面。详细代码如下：

```
<%@ page contentType="text/html; charset=gb2312" language="java" errorPage="error.jsp" %>
```

```
<%@ page import="java.sql.*" %>
<%@ page import="java.text.ParsePosition" %>
<%@ page import="java.text.SimpleDateFormat" %>
<%@ page import="java.util.Date" %>
<%!
    long getDate(String s)
    {
        SimpleDateFormat simpledateformat = new SimpleDateFormat("yyyy-MM-dd");
        ParsePosition parseposition = new ParsePosition(0);
        Date date = simpledateformat.parse(s, parseposition);
        return date.getTime();
    }
%>
<%
String ID = request.getParameter("ID");
String password = request.getParameter("password");
String rpassword = request.getParameter("rpassword");
String name = request.getParameter("name");
String sex = request.getParameter("sex");
String year = request.getParameter("year");
    if(year.length ==1) year = "0"+year;
String mouth = request.getParameter("mouth");
    if(mouth.length == 1) mouth = "0"+mouth;
String day = request.getParameter("day");
    if(day.length == 1) day = "0"+day;
String date = year+"-"+mouth+"-"+day;
String description = request.getParameter("description");

Connection con = null;
PreparedStatement psm = null;
ResultSet rs = null;
try
{
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    String url = "jdbc:mysql://localhost/mydb";
    con = DriverManager.getConnection(url,"root","12345");
    psm = con.prepareStatement("insert into users values(?,?,?,?,?)");
    psm.setString(1,ID);
    psm.setString(2,password);
    psm.setString(3,name);
    psm.setString(4,sex);
    psm.setLong(5,getDate(date));
    psm.setString(6,description);
    psm.executeUpdate();
    session.setAttribute("user",ID);
    response.sendRedirect("index.jsp");
}
catch(Exception e){
    response.sendRedirect("register.jsp");
}
```

```

finally
{
    if(rs!=null){
        try{ rs.close();}catch(Exception e){e.printStackTrace();}
    }
    if(psm!=null){
        try{ psm.close();}catch(Exception e){e.printStackTrace();}
    }
    if(con!=null){
        try{ con.close();}catch(Exception e){e.printStackTrace();}
    }
}
%>

```

代码说明：代码中使用 Java 的声明脚本来定义一个 `getDate(String s)` 方法，以将字符串的日期转换成 `long` 类型的日期（因为数据库表 `users` 中的 `BIRTH` 字段定义的是 `bigint` 类型）。

使用 `request` 内置对象中的 `getParameter("ID")` 方法来获取到上一页面传递过来的表单信息，这里包括用户名、密码（包括重试密码）、姓名、性别、出生年月以及个人描述。

该页面进行数据库插入操作，即将用户填写的个人信息存储到数据库表 `users` 中。如果操作成功，首先调用 `session` 内置对象中的 `setAttribute("user",ID)` 方法将用户 `ID` 保存在 `session` 中，并赋予关键字 `user` 来识别，然后页面跳转到首页；如果操作失败，页面直接返回上一页的注册页面。和所有操作数据库的页面一样，最后一定要在程序中将数据库连接资源释放，详细代码在 `finally{ }` 段中。

12.5.4 创建登录页面login.jsp

该登录页面代码如下：

```

<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*" errorPage="error.jsp"
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<link rel="stylesheet" href="register.css" type="text/css">
<title>用户登陆</title>
<!-- 以下为 javascript 部分，判断用户名和密码栏是否为空-->
<script language="javascript">
<!--
function loginsubmit()
{
    ID = document.form.ID.value;
    password = document.form.password.value;
    if(ID=="||ID.length>25){
        alert("您的用户忘了填写了或者字符数超过规定长度！");
        document.form.ID.focus();
    }
    else if(password=="||password>20||password<6){
        alert("您的密码忘了填写了或者长度不正确！");
        document.form.password.focus();
    }
}

```

```

        else {
            document.form.submit();
        }
    }
-->
</script>
</head>
<body>
<table width="90%" border="0" cellspacing="0" cellpadding="0" align="center">
<tr><td height="35">&nbsp;</td></tr>
<tr><td height="1" align="center">
<%
String info = request.getParameter("info");

if(("1").equals(info))
    out.println("<font size=4 color='red'>用户名不存在，请重新登陆！</font><br><br><br>");
else if(("2").equals(info))
    out.println("<font size=4 color='red'>密码不正确，请重新登陆！</font><br><br><br>");
else
    out.println("<font size=4 color='red'>请登陆！</font><br><br><br>");
%>
</td></tr>
<tr>
<td align="center">
<!--登陆部分-->
<form method=post action="chek_login.jsp" name="form">
<table width="170" border="0" cellspacing="0" cellpadding="0">
<tr>
<td></td>
</tr>
</table>
<table width="170" bgcolor="D2E8FF" style="border-bottom:1px solid #0079CE; border-left:1px solid
#0079CE; border-right:1px solid #0079CE ">
<tr><td height="5"></td></tr>
<tr><td></td><td><input type="text" name="ID" size="13"></td></tr>
<tr><td width="56"></td><td width="113"><input type="Password"
name="password" size="13"></td></tr>
<tr>
<td align="center" colspan="2" height="30">
<a href="javascript:loginsubmit();"></a>&nbsp;  
<a href="register.jsp"></a></td>
</tr>
<tr><td colspan="2" align="right"></td></tr>
</table>
</form>
<!--登陆部分-->
</td>
</tr>
</table>
</body>

```

```
</html>
```

代码说明：该页面代码其实比较简单，将用户填写的用户名和密码提交给 `chek_login.jsp` 页面程序进行处理，并根据返回的操作结果码（1 表示用户名不存在，2 表示密码错误）来适当显示不同提示信息（例如用户名不存在、密码错误等）。该页面的运行效果如图 12.12 所示。

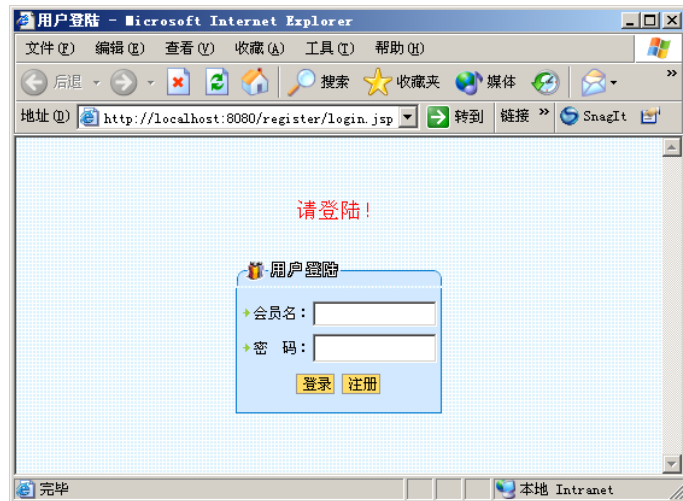


图 12.12 用户登录页面

12.5.5 创建登录处理页面chek_login.jsp

该页面用于处理登录，即进行身份的验证，如果密码错误，则重新返回到登录页面；登录成功，页面跳转到首页 `index.jsp`。该页面详细代码如下：

```
<%@ page contentType="text/html; charset=gb2312" language="java" import="java.sql.*" errorPage="error.jsp"
%>
<%
String ID = request.getParameter("ID");
String password = request.getParameter("password");
String info = "0";
Connection con = null;
Statement sm = null;
ResultSet rs = null;
try
{
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    String url = "jdbc:mysql://localhost/mydb";
    con = DriverManager.getConnection(url,"root","12345");
    sm = con.createStatement();
    rs = sm.executeQuery("select * from users where USER_ID='"+ID+"'");
    if(rs.next()){
        if(rs.getString("password").equals(password)){
            response.sendRedirect("index.jsp");
            session.setAttribute("user",ID);
        }
        else
            response.sendRedirect("login.jsp?info=2");
    }
}
```



```
    }
    else{
        response.sendRedirect("login.jsp?info=1");
    }
}
catch(Exception e){
    e.printStackTrace();
}
finally
{
    if(rs!=null){
        try{ rs.close();}catch(Exception e){e.printStackTrace();}
    }
    if(sm!=null){
        try{ sm.close();}catch(Exception e){e.printStackTrace();}
    }
    if(con!=null){
        try{ con.close();}catch(Exception e){e.printStackTrace();}
    }
}
%>
```

代码说明：用户将登录信息提交到 `chek_login.jsp` 进行验证。首先查找数据库，如果该用户不存在，则返回到 `login.jsp` 页面，并提示“用户不存在”信息；然后验证密码是否正确，如果密码填写正确，返回首页，否则返回到 `login.jsp`，并提示“密码不正确”信息。

12.5.6 创建注销页面logout.jsp

注销页面比较简单，即将用户登录信息从 `session` 中除去，然后页面跳转到登录页面。该页面详细代码如下：

```
<%@ page contentType="text/html; charset=GBK" language="java" import="java.sql.*" errorPage="../error.jsp"
%>
<%
session.removeAttribute("user");
response.sendRedirect("login.jsp");
%>
```

代码说明：使用 `session` 内置对象中的 `removeAttribute()` 方法来去除 `user` 信息，`response` 内置对象中的 `sendRedirect()` 方法来进行页面的跳转。

至此，实例的所有页面代码创建完毕。

12.6 本章小结

本章通过一个完整的实例来演示使用 `Eclipse+Lomboz` 工具来架构 `J2EE` 项目，其中包括 `Eclipse` 和 `Lomboz` 工具的环境配置。学完本章后，读者应该能够熟练使用相关工具，并了解数据库和表的创建。其中功能和数据库设计是学习的重点。