

序号: 24

学号: 16430121



常州大学

实 验 报 告

课 程 名 称: Java EE 程序设计与应用开发

实 验 题 目: JSP 内置对象

实 验 次 数: 第 一 次实验 实验时间: 2019 年 4 月 9 日

学 生 姓 名: 林锦雄

学 院: 信息数理 专 业 班 级: 计算机 162

指 导 教 师: 陆洁茹 评 阅 成 绩:

一、实验目的：

1. 掌握 JSP 技术的基本语法。
2. 掌握 JSP 内置对象的使用方法。
3. 掌握 JSP 开发简单用户登录功能的方法。

二、实验仪器、设备

1、硬件环境

PC 微机；2G 以上内存；VGA 显示格式。

2、软件环境

Windows XP 以上操作系统，JDK，Tomcat 服务器等。

三、实验内容与要求

在初始界面设计一个菜单如图 1，列出本次实验的操作选项，1.1 为第一个题目简易的聊天室，1.2 为一个登录模块，如果用户保存登录信息则下次登陆可以直接进入欢迎界面，1.3 和 1.4 为图书购物设计。菜单设计代码如下：

```
1. <!-- index.jsp -->
2. <html>
3.   <head>
4.     <title>初始菜单</title>
5.   </head>
6.   <body>
7.     <table border="2">
8.       <tr>
9.         <th>编号</th>
10.        <th>题目</th>
11.        <th>操作</th>
12.      </tr>
13.      <tr>
14.        <form action="chatframeset.html" method="post">
15.          <td>1.1</td>
16.          <td>简易聊天室</td>
17.          <td><input type="submit" value="前往"> </td>
18.        </form>
19.      </tr>
20.      <tr>
21.        <form action="login.jsp" method="post">
22.          <td>1.2</td>
23.          <td>登陆模块</td>
24.          <td><input type="submit" value="前往"> </td>
25.        </form>
26.      </tr>
27.      <tr>
28.        <form action="historyBooks.jsp" method="post">
29.          <td>1.3</td>
30.          <td>历史书目</td>
```

```

31.         <td><input type="submit" value="前往"> </td>
32.     </form>
33. </tr>
34. <tr>
35.     <form action="computerBooks.jsp" method="post">
36.         <td>1.4</td>
37.         <td>计算机书目</td>
38.         <td><input type="submit" value="前往"> </td>
39.     </form>
40. </tr>
41. </table>
42. </body>
43. </html>

```

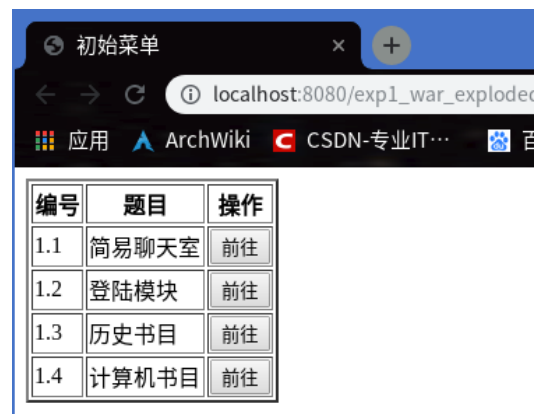


图 1.初始菜单

1. 编写一个网络聊天室

(1) 需求说明

聊天室分上下两部分内容，上面显示最新聊天内容，下面是聊天室发言部分，允许用户将自己的言论发添加到聊天室中。

(2) 设计思路

使用窗口分割，将两个网页显示在同一个窗口中。chatframeset.html。上面的网页显示聊天信息，聊天信息从 application 对象中获取。message.jsp。下面的网页提供表单，用于发送信息，当用户提交信息后，将对用户提交的信息进行验证，然后写到 application 对象中。talk.jsp。

chatframeset.html 文件使用<iframe>标签进行窗口分割，一共分割出两个网页，上面的是 message.jsp 用于显示聊天信息记录，用<hr>标签分隔两个网页，下面的是 talk.jsp，用于提供给用户进行聊天信息输入以及发送，发送的信息会在上面的 message.jsp 中显示出来。如图 2 所示。

```

1. <!-- chatframeset.html -->
2. <iframe src="message.jsp" width="500px" height="200px" scrolling="yes"></iframe>
3. <hr>
4. <iframe src="talk.jsp" width="500px" height="50px"></iframe>

```

message.jsp 用于显示聊天信息，该页面设计为 2 秒刷新一次，每次刷新都会访问一次 application 对象中存储的 message 历史信息，然后依次打印到网页上。如图 3 所示。

```
1. <!-- message.jsp -->
2. <head>
3.     <title>message</title>
4.     <meta http-equiv="refresh" content="2">
5. </head>
6. <body>
7.     <%
8.         String newMessage = (String) application.getAttribute("message");
9.         if(newMessage != null)
10.             out.print(newMessage);
11.     %>
12. </body>
```

talk.jsp 用于提供用户聊天输入的表单，用户把需要输入的聊天信息写到文本框中，点击发送，需要发送的消息就会通过 application 对象保存起来，然后通过 application 对象就会在 message.jsp 中显示出来，消息内容包括了发送人发送的时间和用户地址。如图 3 所示。

```
1. <!-- talk.jsp -->
2. <%
3.     request.setCharacterEncoding("utf-8");
4.     String newMessage = request.getParameter("message");
5.     if(newMessage != null) {
6.         Date date = new Date();
7.         DateFormat df = new SimpleDateFormat("YYYY-MM-DD hh:mm:ss");
8.         newMessage = df.format(date) + "<br>" + request.getRemoteAddr() + "用户:
9.         <br>" + newMessage + "<br>";
10.        String oldMessage = (String) application.getAttribute("message");
11.        if(oldMessage != null) {
12.            newMessage = oldMessage + "<br>" + newMessage;
13.        }
14.        application.setAttribute("message", newMessage);
15.    }
16.    %>
17. <form style="text-align: center" action="talk.jsp" method="post">
18.     <input type="text" name="message"> <input type="submit" value="发送">
19. </form>
```

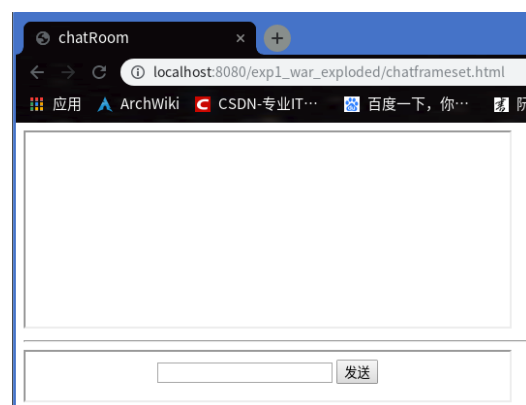


图 2.聊天室初始界面

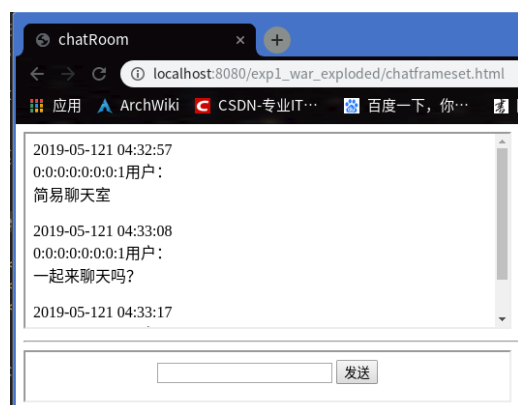


图 3.聊天室使用

2. 编写一个用户登录界面，输入账号和密码，让用户选择“是否保存登录状态”，如果账号密码相等，则登录成功，进入欢迎页面。在登录时，如果保存了登录状态，下次登录时，如果访问登录页面，则进入欢迎页面。但是，客户如果没有保存登录状态，也没有经过登录就访问欢迎页面，则跳转到登录页面。

login.jsp 网页提供了登录表单，表单内容有账号、密码、是否保存登录信息和登录/重置，如图 4 所示。该网页在每次访问时会查询本地 cookies 是否有保存用户登录的账号密码等信息，如果有则可以跳过本次登录操作，并成功登录。

举例输入账号：admin 以及密码：admin，点击保存登录状态并准备进行登录，如图 5 所示。

```
1. <!-- login.jsp -->
2. <%
3.     Cookie[] cookies = request.getCookies();
4.     for(int i = 0; i < cookies.length; i++) {
5.         System.out.println(cookies[i].getName() + ":" + cookies[i].getValue());
6.         if(cookies[i].getName().equals(cookies[i].getValue())) {
7.     %>
8.         <jsp:forward page="welcome.jsp"></jsp:forward>
9. <%
10.         break;
11.     }
12. }
13. %>
14. <form method="post" action="loginResult.jsp">
15.     账号: <input type="text" name="account" value=""><br>
16.     密码: <input type="password" name="passwd" value=""><br>
17.     <input type="radio" name="isKeep" value="true">保存登陆状态<br>
18.     <input type="submit" value="登陆"> ~
19.     <input type="reset" value="重置"><br>
20. </form>
```

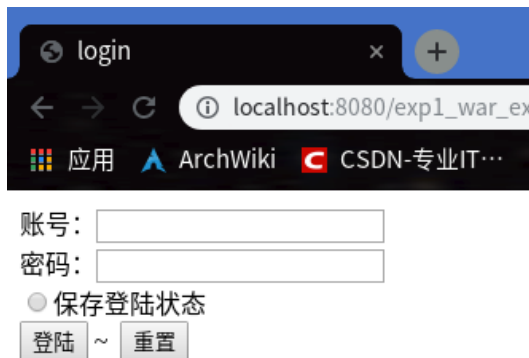


图 4.初始登录界面

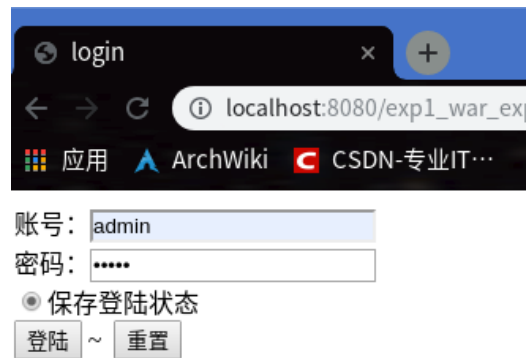


图 5.登录操作

loginResult.jsp 接收 login.jsp 页面传过来的登录表单信息，如果账号和密码相同并且不为空则为登录成功，会跳转到 welcome.jsp，跳转时会附带一个 flag，表示是否从 loginResult.jsp 这个网页跳转到 welcome.jsp；否则为登录失败，打印失败信息。如果用户登录时选择了保存登录状态，则会在此新建一个 cookie 并把用户的登录账号、密码等信息保存在本地。登录成功如图 6 所示。

```
1. <!-- loginResult.jsp -->
```

```

2. <%
3.     boolean isKeep = Boolean.parseBoolean(request.getParameter("isKeep"));
4.     String account = request.getParameter("account");
5.     String password = request.getParameter("passwd");
6.     if(account != null && password != null && !account.equals("") && !password.equals("") && account.equals(password)) {
7.         if(isKeep == true) {
8.             Cookie cookie = new Cookie(account,password);
9.             cookie.setMaxAge(20);
10.            response.addCookie(cookie);
11.            System.out.println(account + ":::::" + password);
12.        }
13.    }%>
14.    <jsp:forward page="welcome.jsp?flag=1"></jsp:forward>
15. <%
16.     } else {
17.         out.print("登陆失败!");
18.     }
19. %>

```

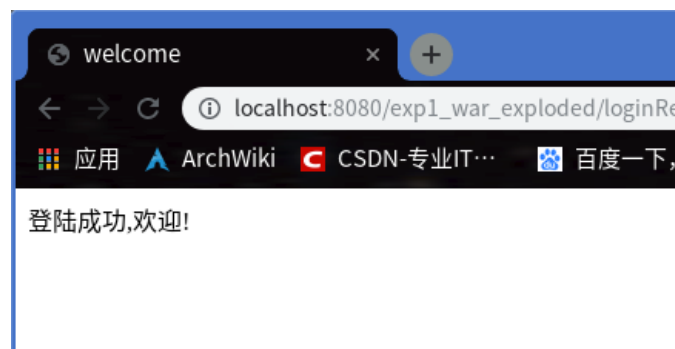


图 6.首次登录成功

已保存过登录状态的情况下直接访问 welcome.jsp 网页，会检查用户是否把登录信息保存在本地 cookie 中，以及判断用户是直接访问 welcome.jsp 还是通过 loginResult.jsp 进行的跳转。

```

1. <!-- welcome.jsp -->
2. <%
3.     String flag = request.getParameter("flag");
4.     String hadKeep = null;
5.     Cookie[] cookies = request.getCookies();
6.     for(int i = 0;i < cookies.length;i++) {
7.         System.out.println(cookies[i].getName() + "====" + cookies[i].getValue());
8.         if(cookies[i].getName().equals(cookies[i].getValue())) {
9.             hadKeep = cookies[i].getName();
10.            break;
11.        }
12.    }
13.    if(hadKeep == null && flag == null) {
14.    }%>
15.    <jsp:forward page="login.jsp"></jsp:forward>
16. <%
17.     }
18. %>
19. 登陆成功,欢迎!

```

如果从 cookie 中查询到有保存的登录信息或者是从 loginResult.jsp 成功登录跳转过来表示允许登录，则打印欢迎信息，如图 7 所示。

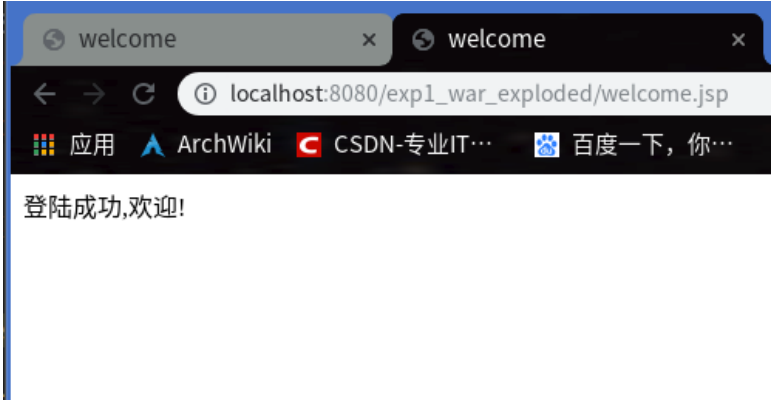


图 7.保存登录信息后直接访问 welcome 网页

如果没有保存好的用户登录信息并且是直接访问，则会判定为非法登录，跳转到登录界面等待用户登录。如图 8 所示。

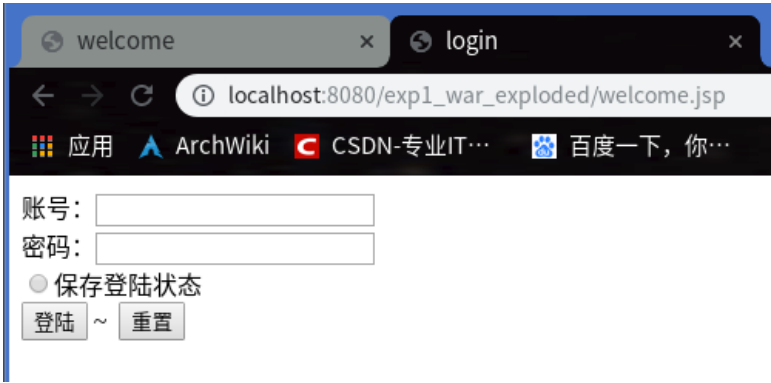


图 8.没保存登录信息直接访问 welcome 网页

3. 编写两个页面，一个显示一些历史图书的名称和价格，一个显示一些计算机书的名称和价格。每本书后面都有一个链接：购买。单击链接，能够将该书加到购物车。每个页面上都有链接：显示购物车。单击该链接，能够显示购物车的内容，每个内容后面都有一个“删除”链接，单击，将该图书从购物车中删除。

historyBooks.jsp 和 computerBooks.jsp 两个网页分别显示一些已保存好的历史书和计算机书，每本书后面都有一个购买链接，单击链接则会把对应的书通过 session 对象加入到购物车 carts 中，网页底部有一个显示购物车链接，点击链接则会跳转到显示购物车的页面并显示出用户添加到购物车的书单。historyBooks.jsp 页面如图 9 所示；computerBooks.jsp 页面如图 10 所示。

```
1. <!-- historyBooks.jsp -->
2. <table border="2" >
3.     <tr>
4.         <th>书名</th>
5.         <th>价格</th>
6.         <th>操作</th>
```

```

7.     </tr>
8.     <tr>
9.         <td>华夏五千年</td>
10.        <td>78.5</td>
11.        <td><a href="historyBooks.jsp?book=华夏五千年">购买</a> </td>
12.    </tr>
13.    <tr>
14.        <td>商鞅变法</td>
15.        <td>43.7</td>
16.        <td><a href="historyBooks.jsp?book=商鞅变法">购买</a> </td>
17.    </tr>
18.    <tr>
19.        <td>贞观之治</td>
20.        <td>66</td>
21.        <td><a href="historyBooks.jsp?book=贞观之治">购买</a> </td>
22.    </tr>
23.    <tr>
24.        <td>改革开放</td>
25.        <td>87.8</td>
26.        <td><a href="historyBooks.jsp?book=改革开放">购买</a> </td>
27.    </tr>
28. </table>
29. <%
30.     ArrayList books = (ArrayList) session.getAttribute("books");
31.     if(books == null) {
32.         books = new ArrayList();
33.         session.setAttribute("books",books);
34.     }
35.     String book = request.getParameter("book");
36.     if(book != null) {
37.         book = new String(book.getBytes("utf-8"));
38.         books.add(book);
39.     }
40. %>
41. <hr>
42. <a href="showCart.jsp">显示购物车</a>

```



图 9.历史书目



图 10.计算机书目


```

1. <!-- computerBooks.jsp -->
2. <table border="2" >
3.     <tr>
4.         <th>书名</th>
5.         <th>价格</th>
6.         <th>操作</th>
7.     </tr>
8.     <tr>
9.         <td>计算机组成原理</td>
10.        <td>48.5</td>
11.        <td><a href="computerBooks.jsp?book=计算机组成原理">购买</a> </td>
12.    </tr>
13.    <tr>
14.        <td>操作系统</td>
15.        <td>33.7</td>
16.        <td><a href="computerBooks.jsp?book=操作系统">购买</a> </td>
17.    </tr>
18.    <tr>
19.        <td>数据结构</td>
20.        <td>57</td>
21.        <td><a href="computerBooks.jsp?book=数据结构">购买</a> </td>
22.    </tr>
23.    <tr>
24.        <td>数据库基础</td>
25.        <td>43.8</td>
26.        <td><a href="computerBooks.jsp?book=数据库基础">购买</a> </td>
27.    </tr>
28.    <tr>
29.        <td>编译原理</td>
30.        <td>73.9</td>
31.        <td><a href="computerBooks.jsp?book=编译原理">购买</a> </td>
32.    </tr>
33. </table>
34. <%
35.     ArrayList books = (ArrayList) session.getAttribute("books");
36.     if(books == null) {
37.         books = new ArrayList();
38.         session.setAttribute("books",books);
39.     }
40.     String book = request.getParameter("book");
41.     if(book != null) {
42.         book = new String(book.getBytes("utf-8"));
43.         books.add(book);
44.     }
45. %>
46. <hr>
47. <a href="showCart.jsp">显示购物车</a>

```

showCart.jsp 为购物车网页，该网站会显示用户想购买的全部书单，每本书后面有一个删除链接，点击链接则会删除从购物车中删除对应的书。我预先在历史书和计算机书中添加了部分书进入购物车，如图 11 所示。删除购物车中的前三本和最后一本书后的购物车如图 12 所示。

```

1. <!-- showCart.jsp -->
2. <table border="2">
3.     <tr>
4.         <th colspan="2">购物车</th>
5.     </tr>
6.     <%
7.         ArrayList books = (ArrayList) session.getAttribute("books");
8.         String book = request.getParameter("delete");
9.         if(book != null) {
10.             books.remove(book);
11.         }
12.         if(books != null) {
13.             for(int i = 0; i < books.size(); i++) {
14.                 %>
15.                 <tr>
16.                     <td><%=books.get(i)%></td>
17.                     <td><a href="showCart.jsp?delete=<%=books.get(i)%>">删除</a> </td>
18.                 </tr>
19.                 <%
20.             }
21.         }
22.     %>
23. </table>
24. <hr>
25. <a href="http://localhost:8080/exp1_war_exploded/">返回</a>

```



图 11.购物车



图 12.删除部分书后的购物车

四、实验心得

本次实验，加深了我对 JSP 各种内置对象的理解和使用，特别是 session、application 和 cookie 对象的应用，这些内置对象大大方便了我们在各种网页之间传递数据以达到各种各样的需求和效果。