



JavaEE 实验一

实验报告

姓　　名　　 Mcrivers

班　　级　　 软件工程2102班

学　　号

提交日期　　 2023.9.12

目录

[一、基础实验——Servlet 与 JSP 基础开发 1](#_Toc151331194)

[1. 运行结果截图和修改后的关键代码 1](#_Toc151331195)

[2.实验收获及总结 3](#_Toc151331196)

[二、提高实验——Servlet 与 JSP 集成的 MVC 方案 3](#_Toc151331197)

[1.运行结果截图 3](#_Toc151331198)

[2. 查找相关资料，总结 DAO 设计模型的优缺点 4](#_Toc151331199)

[3. 实验收获及总结 4](#_Toc151331200)

[三、扩展实验——JDBC 与 DAO 设计模式 5](#_Toc151331201)

[1. 运行结果截图 5](#_Toc151331202)

[2. 总结 DAO 设计模型的优点 7](#_Toc151331203)

[3.实验收获及总结 7](#_Toc151331204)

# 一、基础实验——Servlet 与 JSP 基础开发

## 1. 运行结果截图和修改后的关键代码

**8. 运行 login.jsp 页面，输入用户名和密码并记录运行结果**

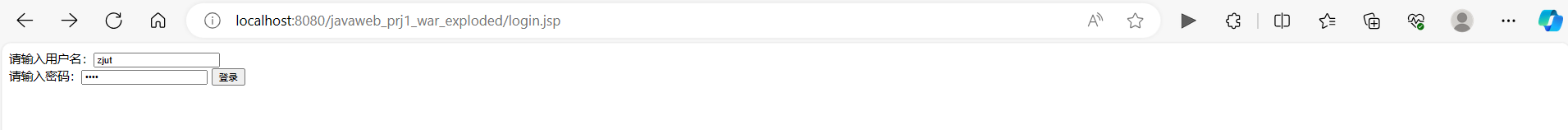


图1.1 login.jsp登录页面

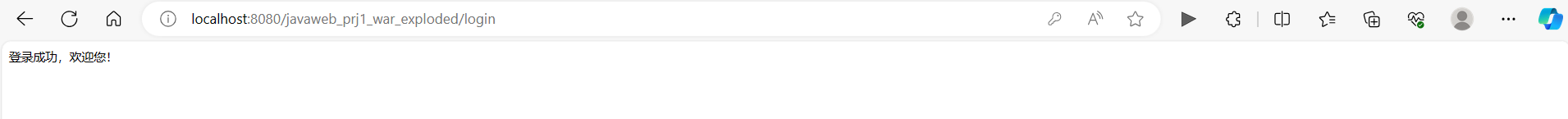


图1.2 登录成功页面（用户名和密码均为zjut）

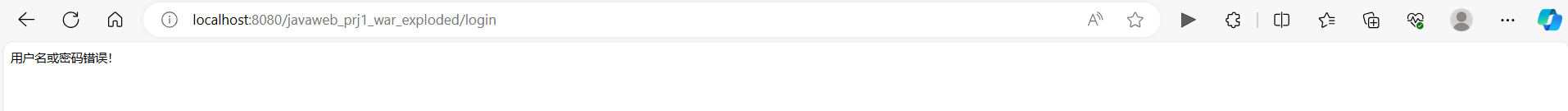


图1.3 登录失败页面

**9. 修改 login.jsp 页面，使用表格对表单域进行对齐排列，运行并观察结果**

观察到，表单域已经成功对齐

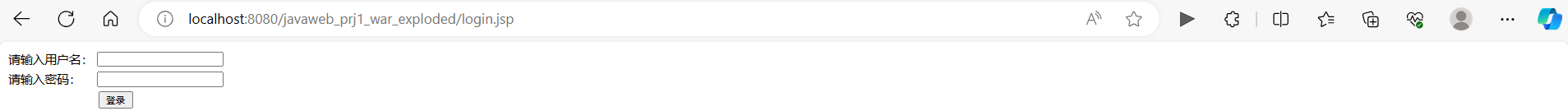


图1.4 login.jsp登录页面

**关键代码：**

<%@ **page** language="java" contentType="text/html; charset=GBK" pageEncoding="GBK"%>  
<!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=GBK">  
 <title>用户登录页面</title>  
</head>  
<body>  
<form action="login" method="post">  
 <table>  
 <tr>  
 <td>请输入用户名：</td>  
 <td><input name="username" type="text"></td>  
 </tr>  
 <tr>  
 <td>请输入密码：</td>  
 <td><input name="password" type="password"></td>  
 </tr>  
 <tr>  
 <td colspan="2" align="center">  
 <input type="submit" value="登录">  
 </td>  
 </tr>  
 </table>  
</form>  
</body>  
</html>

**10. 修改 login.jsp 页面，使用 JavaScript 对用户名表单域 username 和密码表单域 password 进行校验(校验规则：不能为空且不能超过 6 位)**

**观察到**

**关键代码：**

<%@ **page** language="java" contentType="text/html; charset=GBK" pageEncoding="GBK"%>  
<!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=GBK">  
 <title>用户登录页面</title>  
 <script>  
 function validateForm() {  
 var username = ***document***.getElementById("username").value;  
 var password = ***document***.getElementById("password").value;  
  
 if (username === "" || username.length > 6) {  
 alert("用户名不能为空且不能超过 6 位");  
 return false;  
 }  
  
 if (password === "" || password.length > 6) {  
 alert("密码不能为空且不能超过 6 位");  
 return false;  
 }  
  
 return true;  
 }  
 </script>  
</head>  
<body>  
<form action="login" method="post" onsubmit="return validateForm()">  
 <table>  
 <tr>  
 <td>请输入用户名：</td>  
 <td><input id="username" name="username" type="text"></td>  
 </tr>  
 <tr>  
 <td>请输入密码：</td>  
 <td><input id="password" name="password" type="password"></td>  
 </tr>  
 <tr>  
 <td colspan="2" align="center">  
 <input type="submit" value="登录">  
 </td>  
 </tr>  
 </table>  
</form>  
</body>  
</html>



图1.5 login.jsp登录页面

## 2.实验收获及总结

通过这次实验，我很好地回忆起了JavaWeb的开发技能，并对配置方法等技能有了全新的理解，这将对我之后的学习大有裨益。

# 二、提高实验——Servlet 与 JSP 集成的 MVC 方案

## 1.运行结果截图

**6. 运行 login.jsp 页面，输入用户名和密码并运行**

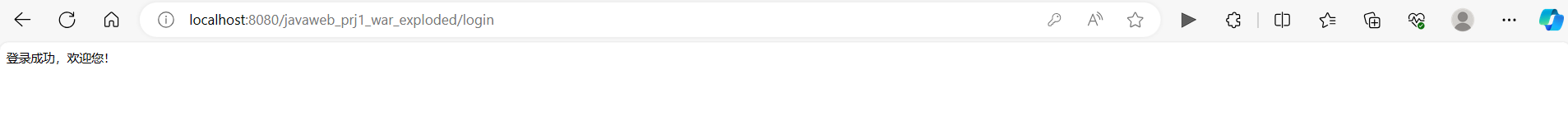


图2.1 登录成功页面（用户名和密码均为zjut）

**8. 运行并观察修改后的结果**

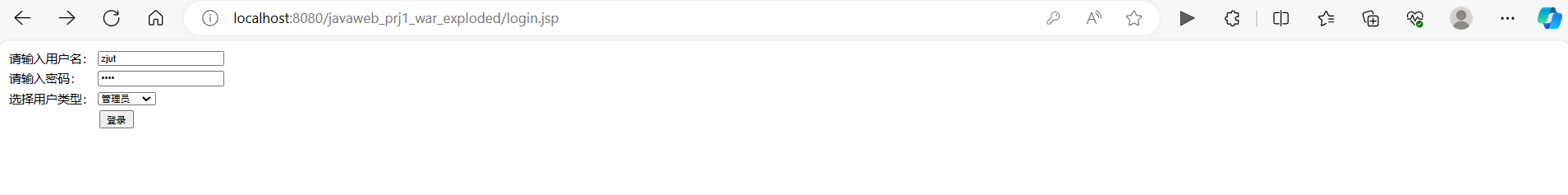


图2.2 登录页面



图2.3 登录成功页面（用户名和密码均为zjut且选择管理员用户）

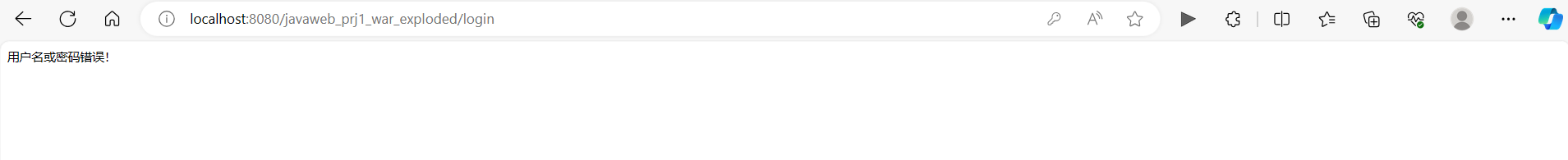


图2.4 登录失败页面

## 2. 查找相关资料，总结 DAO 设计模型的优缺点

**优点：**

①分离关注点

②可维护性好

③可重用性高

④并行开发

⑤易于测试

**缺点：**

①复杂性增加：MVC 架构需要开发人员熟悉并遵循一定的规范，导致整体的复杂性增加。

②过度的耦合：控制器和视图之间可能会存在过度耦合，导致修改一个组件可能会影响其他组件。

## 3. 实验收获及总结

在该实验中，我掌握了 JavaBeans 的编写要点和 MVC 设计模式的核心概念。通过该实验，我深入理解了模型、视图和控制器各自的职责以及它们之间的交互关系。

# 三、扩展实验——JDBC 与 DAO 设计模式

## 1. 运行结果截图

**8. 运行 login.jsp 页面，输入用户名和密码并运行，观察运行结果，如果出现运 行错误，请尝试修正**

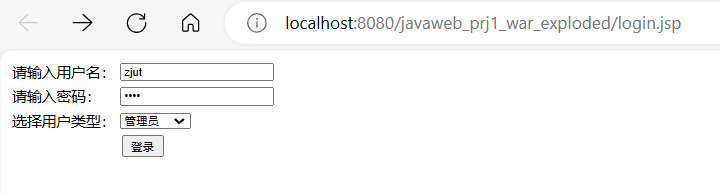


图3.1 登录页面

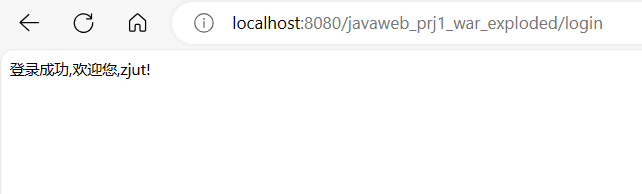


图3.2 登录成功页面

**9. 对 javaweb-prj1 工程进行如下修改**

（1）修改登录成功的条件为用户名、密码和用户类型三者匹配

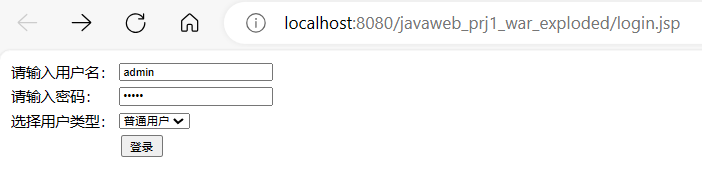


图3.3 登录页面

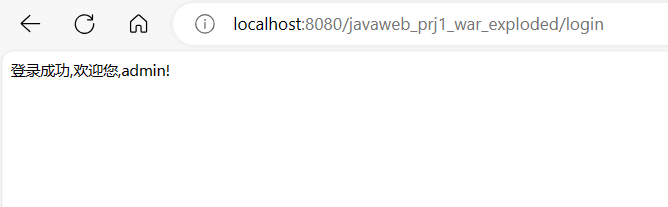


图3.4 登录成功页面

（2）新增一个用户注册视图 register.jsp，用于普通用户的注册



图3.5 注册页面

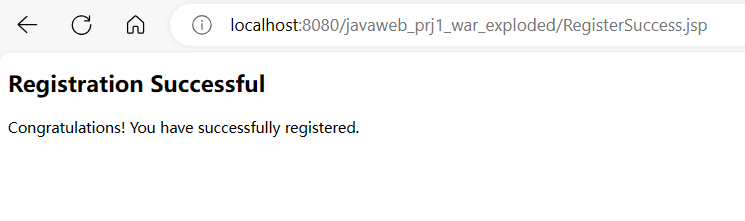


图3.6 注册成功页面

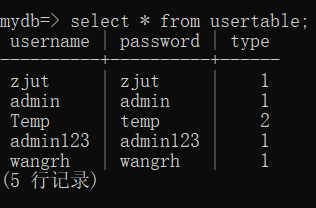


图3.7 查询数据库，发现数据已成功插入

## 2. 总结 DAO 设计模型的优点

①分离数据访问逻辑和业务逻辑： DAO将数据访问逻辑从业务逻辑中分离出来，提高了代码的可维护性和可扩展性。

②提高可维护性： 将数据访问逻辑封装在DAO中使得整个应用程序更易于维护。当需要对数据库进行更改或优化时，只需在DAO层进行修改，而不用修改业务逻辑。

③提供了抽象层： DAO模式通过提供抽象接口隐藏了底层数据库操作的细节，业务逻辑层可以通过DAO接口来访问数据。

④提升安全性： 通过DAO层来控制数据库的访问，可以更好地管理数据库访问权限，避免了直接在业务逻辑中进行数据库访问操作，从而提升了安全性。

⑤增加可测试性： 由于DAO层是接口驱动的，业务逻辑可以通过模拟DAO接口来进行单元测试，不依赖于实际的数据库连接。

## 3.实验收获及总结

在完成这个实验的过程中，我学会了使用SQL语句对数据库进行基本操作，掌握了JDBC连接数据库的基本步骤，了解了DAO设计模式的应用。通过整合MVC和DAO模式，我加深了对模型和实体之间对应关系的理解，同时也意识到业务逻辑与数据访问逻辑分离的优点。