**实验二 JSP表单开发及访问数据库**

**章崇文 计算机222 202202296**

一、实验目的

1．掌握JSP表单的开发方法。

2．熟悉JDBC技术和使用JDBC连接各种数据库。

二、实验学时

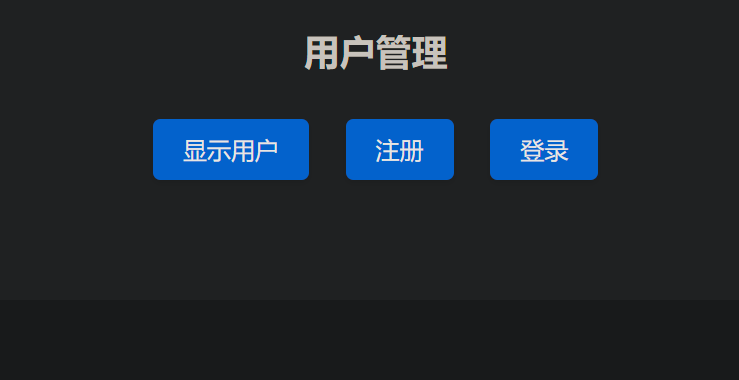
2H

三、实验性质

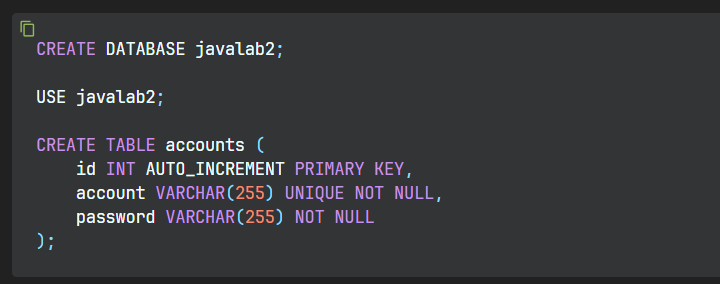
综合性实验

四、实验内容

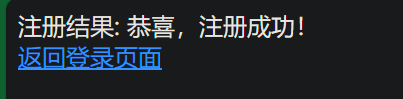
实现注册与登录功能：

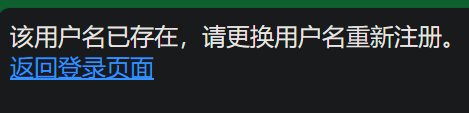


1. 创建一个数据库，在数据库建立用户表。（5分）



2．制作一个注册表单，可以输入账户和密码并提交（在数据提交之前用JS对表单数据进行有效性验证），将表单提交的数据写入数据库。（30分）



情况一 用户注册成功

情况二用户名已经注册

1. 制作一个登录表单，输入账号和密码，通过数据库进行验证，如果账号、密码验证通过，则显示“登录成功”，否则显示“登录失败”。（30分）



登陆界面和登入情况

通过jdbc来进行登陆逻辑的后端

<%

String account = request.getParameter("account");

String password = request.getParameter("password"); // 实际应用中密码应该加密存储

String url = "jdbc:mysql://localhost:3306/javalab2";

Connection con = null;

PreparedStatement checkStmt = null;

PreparedStatement insertStmt = null;

try {

Class.forName("com.mysql.cj.jdbc.Driver");

con = DriverManager.getConnection(url, "root", "123456");

String checkSql = "SELECT \* FROM accounts WHERE account = ?";

checkStmt = con.prepareStatement(checkSql);

checkStmt.setString(1, account);

ResultSet rs = checkStmt.executeQuery();

if (rs.next()) {

out.println("该用户名已存在，请更换用户名重新注册。");

} else {

// 如果账户名不存在，执行插入操作

String insertSql = "INSERT INTO accounts(account, password) VALUES (?, ?)";

insertStmt = con.prepareStatement(insertSql);

insertStmt.setString(1, account);

insertStmt.setString(2, password); // 实际应用中密码应在存储前加密

int n = insertStmt.executeUpdate();

out.println("注册结果: ");

if (n > 0) {

out.println("恭喜，注册成功！");

} else {

out.println("注册失败，请重试。");

}

}

} catch (Exception e) {

out.println("注册过程中出现错误：" + e.getMessage());

} finally {

try {

if (checkStmt != null) checkStmt.close();

if (insertStmt != null) insertStmt.close();

if (con != null) con.close();

} catch (Exception e) {}

}%>

4．从数据库中读取所有的用户数据，并在页面中显示。（25分）



五、实验总结

请书写你对本次实验有哪些实质性的收获和体会，以及对本次实验有何良好的建议？（10分）

JSP表单开发能力增强：通过亲手实现注册与登录功能，我不仅掌握了JSP表单的基本语法和标签使用，如<form>、<input>、<button>等，还学习到了如何利用JavaScript进行前端表单验证，保证用户输入的有效性和安全性。这种前后端结合的实践加深了我对Web开发流程的理解。

JDBC技术的应用：实验中，我亲自配置了数据库连接，并通过JDBC执行SQL语句来实现数据的插入、查询等操作。这个过程让我明白了如何在Java应用中与数据库交互，包括数据库驱动的加载、Connection对象的获取、PreparedStatement的使用以及结果集的处理。这种动手操作极大地增强了我解决实际问题的能力。

数据库设计与管理：在创建用户表的过程中，我学习了如何设计合理的数据库表结构，包括字段的选择、数据类型的设定以及约束条件的添加。这对于保证数据的一致性和完整性至关重要。同时，通过实际操作数据库，我也意识到了数据管理与维护的重要性。

安全性意识提升：在实现用户注册与登录功能时，我特别注意了密码的处理，了解了密码加密存储的重要性，虽然实验中可能简化了这一环节，但它提醒我在真实项目中必须采取加密措施保护用户信息安全。

针对本次实验，我有一些建议：

增加安全性实践：实验中可以进一步引导学生学习并实践密码的加密存储（如使用BCrypt或SHA-256算法），以及如何防范SQL注入等安全威胁，使学生在开发过程中养成良好的安全编程习惯。

数据库操作的异常处理：虽然实验主要关注功能实现，但在实际开发中，数据库操作常常伴随异常情况，如连接失败、查询错误等。建议增加异常处理的教学内容，让学生学会如何编写健壮的数据库交互代码。

实战项目演练：如果条件允许，可以设计一个完整的Web应用项目，让学生从需求分析、数据库设计、功能实现到部署上线全程参与，这样不仅能够综合运用所学知识，还能提升项目管理和团队协作能力。

总之，这次实验是一次非常宝贵的学习经历，不仅巩固了我的理论知识，更重要的是提高了我的实战技能，为将来从事Web开发工作打下了坚实的基础。