**实验报告**

**学院（系）名称：**计算机科学与工程学院

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **姓名** | | 王帆 | | **学号** | 20152180 | | **专业** | 计算机科学与技术 |
| **班级** | | 2015级1班 | | **实验**  **项目** | JSP、Servlet及JDBC应用开发 | | | |
| **课程名称** | | | | Web应用程序设计与开发 | | | **课程**  **代码** | 0668026 |
| **实验时间** | | | | 2018年5月22日 中午5、6节 | | | **实验**  **地点** | 7-215 |
| 考核标 准 | 实验准备（实验目的/工具熟悉情况）10分 | | 实验过程（实验方案可行性及步骤完整性）40分 | 实验报告(实验内容丰富度与格式清晰度) 30分 | | 实验结果（结论正确性以及分析合理性）20分 | **成绩** |  |
|  |  | |  |  | |  | **教师签字：** | |
| 考核内容 | 评价实验目的是否明确，实验工具是否清晰了解以及熟悉情况 | | ○可行，完整  ○可行，不完整  ○不可行，不完整 | ○丰富，清晰  ○较丰富，较清晰  ○丰富，不清晰  ○不丰富，不清晰 | | ○结论正确，分析合理  ○结论正确，分析不充分  ○结论不正确，分析不合理 |
| **一、实验目的**  掌握使用JDBC访问数据库，学会使用JSP的内置对象的使用，学会Servlet和JavaBean等技术的应用。  **二、实验环境**  Windows操作系统，Tomcat，MyEclipse，HBuilder，记事本。  **三、 实验要求**  1.实现系统的用户登录，要求：  （1）创建登录表，并利用此表实现系统的登录功能；  （2）登录成功要求能够保存或维护用户信息，以便能够被其他界面使用用户信息。  2.实现对单表的信息的添加功能，要求：  （1）建立数据库的表结构，至少包含4个以上字段；  （2）利用表单的提交功能。  （3）利用Servlet处理，充分考虑编码及封装。  3.对数据库的表的信息查询功能，要求：  （1）利用JavaBean实现，并将表中的信息按行进行显示。  （2）建立数据库的表结构，至少包含4个以上字段；表的信息可以是学生信息、图书信息、商品信息等。  （3）实现时充分考虑代码的封装。  **四、实验过程记录（源程序、测试用例、测试结果及心得体会等）**  1.实现系统的用户登录  （1）创建登录表，并利用此表实现系统的登录功能；。  表结构如下：  表1 数据表结构（用户信息）   |  |  | | --- | --- | | 名称 | 数据类型 | | userid | INT | | username | VARCHAR | | password | VARCHAR |   （2）登录成功要求能够保存或维护用户信息，以便能够被其他界面使用用户信息。  在Servlet中使用Session对象对登录状态进行存储，以便在其他界面调用验证  后端代码：  response.setContentType("text/html;charset=utf-8");  String username = WebUtil.*getParameter*(request,"username");  String password = WebUtil.*getParameter*(request,"password");  PrintWriter out = response.getWriter();  HashMap<String, String> hashMap = **new** HashMap<String, String>();  hashMap.put("username", username);  hashMap.put("password", password);  **if**(DBUtil.getDataCount("user",hashMap)==1) {  HttpSession session = request.getSession();  session.setAttribute("username",username); //用户名  session.setAttribute("loginState","1"); //登录状态  response.sendRedirect("./loginCheck.jsp");  }  **else** {  HttpSession session = request.getSession();  session.setAttribute("loginState","0"); //登录状态  response.sendRedirect("./loginCheck.jsp");  }  演示：    图1-1 登录前提示并自动跳转    图1-2 登录页面  2.实现对单表的信息的添加功能  建立数据库的表结构，如下：  表2 数据表结构（学生信息）   |  |  | | --- | --- | | 名称 | 数据类型 | | sno | INT | | sname | VARCHAR | | age | VARCHAR | | politicalstate | VARCHAR | | birthday | VARCHAR | | address | VARCHAR | | phone | VARCHAR | | institute | VARCHAR | | demo | VARCHAR |   表单录入情况演示：    图2 表单录入  3.对数据库的表的信息查询功能  前端代码：  <form action="${pageContext.request.contextPath}/FormServlet" method="post">  <table style="width: auto;" class=".table-bordered table table-striped table-hover table-bordered table table-condensed">  <tr>  <th>Key</th>  <th>Value</th>  </tr>  <tr>  <td>sno</td>  <td>  <input type="text" name="sno" id="sno">  </td>  </tr>  <tr>  <td>sname</td>  <td>  <input type="text" name="sname" id="sname">  </td>  </tr>  <tr>  <td>age</td>  <td>  <input type="text" name="age" id="age">  </td>  </tr>  <tr>  <td>politicalstate</td>  <td>  <input type="text" name="politicalstate" id="politicalstate">  </td>  </tr>  <tr>  <td>birthday</td>  <td>  <input type="text" name="birthday" id="birthday">  </td>  </tr>  <tr>  <td>address</td>  <td>  <input type="text" name="address" id="address">  </td>  </tr>  <tr>  <td>phone</td>  <td>  <input type="text" name="phone" id="phone">  </td>  </tr>  <tr>  <td>institute</td>  <td>  <input type="text" name="institute" id="institute">  </td>  </tr>  <tr>  <td>demo</td>  <td>  <input type="text" name="demo" id="demo">  </td>  </tr>  </table>  <div class="form-actions">  <button type="submit" class="btn btn-primary">提交</button>  <button type="button" class="btn">重置</button>  </div>  </form>  后端代码：  request.setCharacterEncoding("UTF-8");  HttpSession session = request.getSession();  String op = request.getParameter("op");  System.out.print(op);  **switch** (op) {  **case** "2Bean":  response.setContentType("text/html;charset=utf-8");    Student student\_bean = **new** Student();  student\_bean.setAddress(request.getParameter("address"));  student\_bean.setAge(request.getParameter("age"));  student\_bean.setBirthday(request.getParameter("birthday"));  student\_bean.setDemo(request.getParameter("demo"));  student\_bean.setInstitute(request.getParameter("institute"));  student\_bean.setPhone(request.getParameter("phone"));  student\_bean.setSname(request.getParameter("sname"));  student\_bean.setSno(Integer.parseInt(request.getParameter("sno")));  student\_bean.setPoliticalstate(request.getParameter("politicalstate"));  session.setAttribute("studentBean",student\_bean);  response.sendRedirect("./FormJavabean.jsp");  **break**;  **case** "2DB":  System.out.println("OK");  response.setContentType("text/html;charset=utf-8");    Student student\_db = **new** Student();  System.out.println("OK2");  student\_db.setAddress(request.getParameter("address"));  student\_db.setAge(request.getParameter("age"));  student\_db.setBirthday(request.getParameter("birthday"));  student\_db.setDemo(request.getParameter("demo"));  student\_db.setInstitute(request.getParameter("institute"));  student\_db.setPhone(request.getParameter("phone"));  student\_db.setSname(request.getParameter("sname"));  student\_db.setSno(Integer.parseInt(request.getParameter("sno")));  student\_db.setPoliticalstate(request.getParameter("politicalstate"));    String sql = "INSERT INTO student(sno,sname,age,politicalstate,birthday,address,phone,institute,demo) VALUES('";  sql+=student\_db.getSno()+"','"+student\_db.getSname()+"','"+student\_db.getAge()+"','"+student\_db.getPoliticalstate()+"','"+student\_db.getBirthday()+"','"+student\_db.getAddress()+"','"+student\_db.getPhone()+"','"+student\_db.getInstitute()+"','"+student\_db.getDemo()+"')";  **if**(DBUtil.executeBatch(sql)){  PrintWriter outPrintWriter = response.getWriter();  outPrintWriter.println("表更新成功！");  System.out.println("表更新成功！");  }  **else**{  PrintWriter outPrintWriter = response.getWriter();  outPrintWriter.println("表更新失败！");  System.out.println("表更新失败！");  }  **break**;  **default**:  session.setAttribute("error","参数错误");  response.sendRedirect("./errorHandle.jsp");  **break**;  }  **private** **void** insertStudentInfo(HttpServletResponse response, String sno) {  **try** {  String string = "";  JSONObject jsonObject = **new** JSONObject();  ArrayList<String> List = **new** ArrayList<>();  String delete = "INSERT INTO student VALUES(?,?,?,?,?,?,?,?,?)";  List.add(delete);  **boolean** result = DBUtil.*executeBatch*(List);  PrintWriter out = response.getWriter();  response.setCharacterEncoding("utf-8");  **if**(result){  jsonObject.put("ret", "1");  }  **else** {  jsonObject.put("ret", "0");  jsonObject.put("reason", "数据库操作失败");  }  out.print(jsonObject);  out.close();  } **catch** (Exception ex) {  ex.printStackTrace();  }  }  演示：    图2-2 表单查询    图2-3 表单更新成功提醒  心得体会：  通过本次实验，我加深了对JSP/Servlet以及JDBC开发的理解。JSP全名为Java Server Pages，是一个简化的Servlet设计。Servlet（Server Applet）是Java Servlet的简称，称为小服务程序或服务连接器，主要功能在于交互式地浏览和修改数据，生成动态Web内容。JDBC（Java DataBase Connectivity,java数据库连接）是一种用于执行SQL语句的Java API，可以为多种关系数据库提供统一访问，它由一组用Java语言编写的类和接口组成。通过对以上技术的运用与实践，我能够初步实现基本WEB应用，并实现与数据库的链接。在今后的学习中，我将继续对WEB应用程序进行了解与分析，实现更多功能。 | | | | | | | | |