

中国矿业大学计算机学院

2017 级本科生课程报告

课程名称	Web 技术
报告时间	2020.10.21
学生姓名	李治远
学 号	07172757
专 业	计算机科学与技术
任课教师	杨勇

《Web 技术》评分表

编号	毕业要求	课程教学目标	考察方式与考察点	占比	得分
1	1.3	目标 1: 掌握 Web 应用程序的工作原理, CSS3 的语法, 应用 CSS3+DIV 技术实现页面美化与布局, JavaScript 内置核心语言对象以及基本语法 JSP 页面的基本构成, 掌握 Servlet 的工作原理、技术特点、分类等基础的 Web 应用程序开发的基础知识, 了解通过 Web 技术解决复杂工程问题的基本方法。	1)针对具体要求, 运用 HTML 、 CSS 、 JavaScript、JSP 内置对象设计并开发静态页面和动态页面。 2)能够针对实际问题, 合理地分析和选择 Web 前端技术与 Web 服务器端开发技术 (JSP、Servlet) 进行 Web 系统设计。	50%	
2	3.4	目标 2: 熟悉目前几种主流的 Web 开发技术, 掌握 4 种内置对象属性范围的区别与应用原则, 掌握 MVCII 模式的原理与特点, 使用该模式进行 Web 应用程序开发。能够针对具体复杂应用问题, 在开发平台上设计实施方案, 完成程序的编码、调试和运行, 并能对结果进行初步的分析和评价。	1)能够针对实际应用问题, 合理地分析和选择 Html, 第三方 JavaScript 库解决客户端设计问题, 选择合理的开发模式进行 Web 系统开发。 2)结合具体较为复杂工程需求, 设计与编程实现、调试, 并能对结果进行初步分析与评价。	50%	
总分				100%	

评阅人:

日 期:

摘 要

在本次 Web 应用开发课程设计中，进行了 3 个实验。

➤ 实验一，静态 Web 页面设计。

做出了“矿大资助中心”专题网站。首页使用 CSS 和 DOM 操作实现的轮播图是实验一的一大特色。实验一重点关注在 HTML、CSS 和 JavaScript 的使用，利用 CSS 对网页进行布局 and 美化，利用 JavaScript 为注册界面实现了简单验证，同时为登录成功与注册成功页面实现跳转。

➤ 实验二，动态 Web 界面设计。

在实验一的基础上，将部分静态的 html 页面转为动态的 JSP 页面，利用 JavaBean 并实现服务器端的验证，防止错误数据的录入。学习 JSP 的一些内置对象，如：request、response 与 session 等，并且利用 session 来维持登录状态和进行注销。也利用了 req 与 resp 设计客户端与服务端的跳转。

实验三，Web 数据库实验。

实现了 JDBC 访问数据库，Filter 过滤器对权限控制和乱码处理。同时，采用 DAO 设计模式对 MySQL 中的数据表进行了增删改查操作。

关键词：轮播图；DOM；JavaBean；JDBC；Servlet；CRUD

目录

1	静态 Web 页面设计	2
1.1	实验说明	2
1.1.1	实验目的	2
1.1.2	实验要求	2
1.2	实验内容	2
1.2.1	页面主体布局	2
1.2.2	登录页	3
1.2.3	首页	5
1.2.4	个人信息页	9
1.2.5	资助金信息查询页	11
1.3	其他展示	13
1.3.1	JavaScript 验证登录	13
1.3.2	JavaScript 实现轮播图	15
1.4	设计心得	17
2	动态 Web 页面设计实验	18
2.1	实验说明	18
2.1.1	实验目的	18
2.1.2	实验要求	18
2.2	实验内容	18
2.2.1	网站结构	18
2.2.2	JavaBean 验证	18
2.2.3	Request 对象	21
2.2.4	Response 对象	23
2.2.5	Session 对象	23
2.2.6	客户端跳转与服务器端跳转	24
2.3	设计心得	25
3	Web 数据库开发实验	26
3.1	实验说明	26
3.1.1	实验目的	26
3.1.2	实验要求	26
3.1.3	实验内容	26
3.2	详细设计与编码	26
3.2.1	设计数据库表格	26
3.2.2	连接数据库	27
3.2.3	Filter 登录权限控制机制	28
3.2.4	数据库连接与关闭公共方法	29
3.2.5	用户表增查改	30
3.2.6	奖助金信息查增删	33
3.3	其他展示	37
3.3.1	角色访问控制	37
3.4	设计心得	38

1 静态 Web 页面设计

1.1 实验说明

1.1.1 实验目的

- 1、配置 Web(TOMCAT)服务器，了解 Web 工作原理。
- 2、熟悉常用 HTML5 标记的含义，能够熟练使用这些标记设计静态 Web 页面，实现 Web 页面上各种元素的合理布局，如表单、表格、图片以及框架等标记的使用。
- 3、了解 CSS 样式表的定义和使用方法，能够使用 CSS 美化和布局 Web 页面。
- 4、掌握 JavaScript 脚本语言的基本语法。
- 5、能够使用 JavaScript 与浏览器对象进行交互。
- 6、能够使用 JavaScript 处理表单和表单元素事件。

1.1.2 实验要求

- 1、使用 HTML5 开发 Web 静态页面。按照 HTML5 的规范设计与开发网站。
- 2、练习 HTML5 的新 HTML5 新特性和效果。
- 3、练习使用 Web 页面开发工具（MyEclipse、Dreamwear、VS.NET 或其它）。
- 4、练习第三方 JavaScript 库的使用。
- 5、完成实验报告和实验成果。

1.2 实验内容

实验主题为“矿大资助中心”网站，包括首页、登录、注册页、个人信息页面与资助金查询页等。其中，使用 css 美化 html，并且实现一些特效，使用 JavaScript 操作 DOM 完成轮播图效果和表单提交信息的验证。

1.2.1 页面主体布局



图表 1-1 页面布局

1.2.2 登录页

登录页 login.html 代码如下所示：

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.     <meta charset="UTF-8">
5.     <title>登录</title>
6.     <link type="text/css" href="static/css/main.css" rel="stylesheet">
7.     <link type="text/css" href="static/css/login.css" rel="stylesheet">
8. </head>
9. <body style="background-image: url('static/images/bg.jpg')">
10. <div class="login">
11.     <div class="title">登录</div>
12.     <form action="index.html" method="post" class="login-
    form" name="loginForm">
13.         <div class="form-item">
14.             <label class="form-label" for="username">用户名</label>
15.             <input class="form-
    input" id="username" name="username" value="07172757" required placeholder="
    用户名">
16.         </div>
17.         <div class="form-item">
18.             <label class="form-label" for="password">密码</label>
19.             <input class="form-
    input" id="password" name="password" value="173510" type="password" required
20.                 placeholder="密码">
21.         </div>
22.         <div class="form-item" style="justify-content: flex-start">
23.             <a href="register.html">没有账号？立即注册</a>
24.         </div>
25.         <button class="submit-btn" type="submit">登录</button>
26.     </form>
27. </div>
28. </body>
29. <script type="text/javascript" src="utils/valid.js"></script>
30. <script type="text/javascript" src="static/js/login.js"></script>
31. </html>
```

登录页 login.css 如下所示：

```
1. .login {
2.     background-color: gray;
```

```

3.     width: 500px;
4.     position: absolute;
5.     left: 50%;
6.     top: 30%;
7.     transform: translate(-50%, -50%);
8.     border-radius: 20px;
9.     opacity: 0.8;
10. }
11. .login .title {
12.     font-size: 32px;
13.     text-align: center;
14.     color: #fff;
15.     margin: 10px auto;
16. }
17. .login .login-form {
18.     display: flex;
19.     flex-direction: column;
20.     justify-content: center;
21.     align-items: center;
22. }
23. .login .login-form .form-item {
24.     width: 400px;
25.     margin: 10px 0;
26.     display: flex;
27.     justify-content: center;
28.     align-items: center;
29. }
30. .login .login-form .form-item .form-label {
31.     color: #fff;
32.     display: inline-block;
33.     width: 80px;
34. }
35. .login .login-form .form-item .form-input {
36.     width: 100%;
37.     height: 40px;
38.     padding: 0 10px;
39.     outline: 0;
40.     border: none;
41.     border-radius: 8px;
42.     flex: 1;
43.     font-size: 16px;
44. }
45. .login .login-form .submit-btn{
46.     margin: 10px 0;

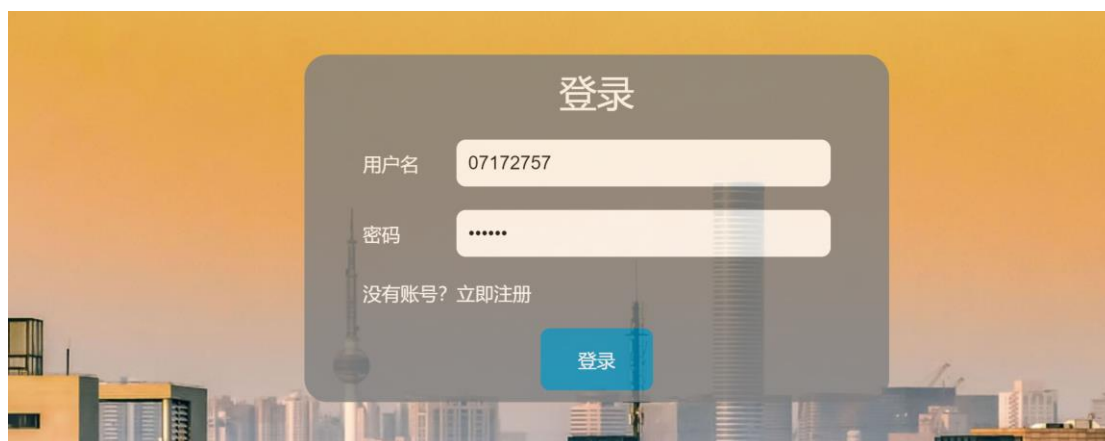
```

```

47.     background-color: #008CBA;
48.     border: none;
49.     color: white;
50.     padding: 16px 32px;
51.     text-align: center;
52.     text-decoration: none;
53.     display: inline-block;
54.     font-size: 16px;
55.     border-radius: 8px;
56.     cursor: pointer;
57. }
58. .login .login-form .submit-btn:hover {
59.     background-color: #006e92;
60. }

```

登录页效果图如下：



图表 1-2 登录页

注册页与登录页类似，不在赘述。

1.2.3 首页

首页 index.html 代码如下所示：

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.     <meta charset="UTF-8">
5.     <title>主页</title>
6.     <link type="text/css" href="static/css/main.css" rel="stylesheet">
7.     <link type="text/css" href="static/css/header.css" rel="stylesheet">
8.     <link type="text/css" href="static/css/banner.css" rel="stylesheet">
9.     <link type="text/css" href="static/css/info.css" rel="stylesheet">
10. </head>
11. <body>
12. <!--顶部导航-->

```



```

13. <div class="navbar">
14.     <div class="container">
15.         <div class="logo-header">
16.             
17.             <div class="title">矿大智慧资助</div>
18.         </div>
19.         <div class="navbar-tool">
20.             <a href="index.html">首页</a>
21.             <a href="self.html">个人信息</a>
22.             <a href="login.html">奖助金查询</a>
23.             <a href="login.html">登录</a>
24.         </div>
25.     </div>
26. </div>
27. <!--轮播图-->
28. <div class="banner">
29.     <div class="images">
30.         
31.         
32.         
33.     </div>
34.     <div class="tools">
35.         <div class="btns">
36.             <div class="last-btn" onclick="lastImage()"><</div>
37.             <div class="next-btn" onclick="nextImage()">>>/div>
38.         </div>
39.         <div class="dots" id="dots">
40.             <div class="dots-
item activate" onclick="activate(0)"><span>1</span></div>
41.             <div class="dots-
item deactivate" onclick="activate(1)"><span>2</span></div>
42.             <div class="dots-
item deactivate" onclick="activate(2)"><span>3</span></div>
43.         </div>
44.     </div>
45. </div>
46. <!--信息聚合-->
47. <div class="info-container">
48. </div>
49. <!--底部-->
50. <div style="width: 1140px;margin:0 auto;text-
align: center;padding: 10px;">
51.     <div>
52.         Author: 计科 17-01 班 李治远

```

```

53.     </div>
54.     <div>
55.         地址：江苏省徐州市大学路 1 号中国矿业大学南湖校区 邮编：221116
56.     </div>
57. </div>
58. </body>
59. <script type="text/javascript" src="static/js/banner.js"></script>
60. </html>

```

首页顶部 header.css 代码如下所示：

```

1. .navbar{
2.     /*position: fixed;*/
3.     z-index: 99;
4.     background-color: #2C2C32;
5.     height: 80px;
6.     width: 100vw;
7.     display: flex;
8.     flex-direction: row;
9.     justify-content: center;
10.    align-items: center;
11.    color: #fff;
12. }
13. .navbar .container{
14.     width: 1170px;
15.     min-width: 1170px;
16.     display: flex;
17.     flex-direction: row;
18.     justify-content: space-between;
19.     align-items: center;
20. }
21. .navbar .container .logo-header{
22.     display: flex;
23.     flex-direction: row;
24.     justify-content: space-between;
25.     align-items: center;
26. }
27. .navbar .container .logo-header .logo{
28.     height: 60px;
29.     padding-right: 20px;
30.     margin: 10px 20px 10px 10px;
31.     border-right: 2px solid #eeeeee;
32. }
33. .navbar .container .logo-header .title{
34.     font-size: 32px;

```

```

35. }
36. .navbar .container .navbar-tool a{
37.     margin: 0 5px;
38. }

```

首页轮播图 banner.css 如下所示:

```

1. .banner {
2.     width: 1170px;
3.     min-width: 1170px;
4.     margin: 10px auto;
5.     position: relative;
6. }
7. .banner .images {
8.     display: flex;
9.     overflow: hidden;
10. }
11. .banner img {
12.     width: 1170px;
13.     min-width: 1170px;
14. }
15. .banner .tools .btns {
16.     width: 1170px;
17.     position: absolute;
18.     top: 200px;
19.     display: flex;
20.     flex-direction: row;
21.     justify-content: space-between;
22. }
23. .banner .tools .btns div {
24.     background-color: #eeeeee;
25.     color: #cacaca;
26.     padding: 16px;
27.     text-align: center;
28.     font-size: 24px;
29.     cursor: pointer;
30.     opacity: 0.5;
31. }
32. .banner .tools .btns div:hover {
33.     color: #a5a5a5;
34.     opacity: 0.9;
35. }
36. #dots {

```

```

37.     position: absolute;
38.     right: 10px;
39.     bottom: 10px;
40.     display: flex;
41.     flex-direction: row;
42.     justify-content: flex-end;
43. }
44.
45. #dots .dots-item {
46.     width: 20px;
47.     height: 20px;
48.     margin-left: 10px;
49.     padding: 10px;
50.     text-align: center;
51.     font-size: 20px;
52.     font-weight: bold;
53.     display: flex;
54.     flex-direction: row;
55.     justify-content: center;
56.     align-items: center;
57.     cursor: pointer;
58.     opacity: 0.5;
59. }
60. #dots .dots-item:hover {
61.     opacity: 0.9;
62. }
63. .activate {
64.     color: #fff;
65.     background-color: #f13b33;
66. }
67. .deactivate {
68.     color: #cacaca;
69.     background-color: #eeeeee;
70. }
71. .deactivate:hover {
72.     color: #a5a5a5;
73. }

```

首页效果图如下图所示：



图表 1-3 首页

1.2.4 个人信息页

个人信息页 self.html 如下所示：

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.     <meta charset="UTF-8">
5.     <title>个人信息</title>
6.     <link type="text/css" href="static/css/main.css" rel="stylesheet">
7.     <link type="text/css" href="static/css/header.css" rel="stylesheet">
8.     <link type="text/css" href="static/css/self.css" rel="stylesheet">
9. </head>
10. <body>
11. <!--顶部导航-->
12. <div class="navbar">
13.     <div class="container">
14.         <div class="logo-header">
15.             
16.             <div class="title">矿大智慧资助</div>
17.         </div>
18.         <div class="navbar-tool">
19.             <a href="index.html">首页</a>
20.             <a href="self.html">个人信息</a>
21.             <a href="login.html">奖助学金查询</a>
```

```

22.         <a href="login.html">登录</a>
23.     </div>
24. </div>
25. </div>
26. <div class="self">
27.     <table class="self-table">
28.         <tr>
29.             <td>姓名: 李治远</td>
30.             <td>学号: 07172757</td>
31.         </tr>
32.         <tr>
33.             <td>学院: 计算机科学与技术学院</td>
34.             <td>年级: 2017</td>
35.         </tr>
36.         <tr>
37.             <td>专业: 计算机科学与技术</td>
38.             <td>班级: 计算机科学与技术 2017-01 班</td>
39.         </tr>
40.     </table>
41. </div>
42. </body>
43. <script type="text/javascript" src="static/js/banner.js"></script>
44. </html>

```

个人信息页 self.css 如下所示:

```

1. .self {
2.     width: 800px;
3.     min-width: 800px;
4.     margin: 10% auto 0;
5.     padding: 2% 5%;
6.     border-radius: 10px;
7.     background-color: #fff;
8. }
9.
10. .self .self-table {
11.     width: 80%;
12.     margin: 0 auto;
13. }

```

个人信息页效果图如下所示:

姓名: 李治远	学号: 07172757
学院: 计算机科学与技术学院	年级: 2017
专业: 计算机科学与技术	班级: 计算机科学与技术2017-01班

图表 1-4 个人信息页

1.2.5 资助金信息查询页

资助金信息 funding.html 如下所示（使用了 bootstrap 插件）：

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.     <meta charset="UTF-8">
5.     <title>资助金信息</title>
6.     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.
7.         integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYxXfFc+NcPb1dKGj7Sk" crossorigin="anonymous">
8.     <link type="text/css" href="static/css/main.css" rel="stylesheet">
9.     <link type="text/css" href="static/css/header.css" rel="stylesheet">
10.    <style>
11.        .funding-container {
12.            width: 1140px;
13.            min-width: 1140px;
14.            margin: 20px auto;
15.        }
16.    </style>
17. </head>
18. <body>
19. <!-- 顶部导航 -->
20. <div class="navbar">
21.     <div class="container">
22.         <div class="logo-header">
23.             
24.             <div class="title">矿大智慧资助</div>
25.         </div>
26.         <div class="navbar-tool">
27.             <a href="index.html">首页</a>
28.             <a href="self.html">个人信息</a>
29.             <a href="login.html">奖助金查询</a>
30.             <a href="login.html">登录</a>
```

```

31.         </div>
32.     </div>
33. </div>
34. <div class="funding-container">
35.     <table class="table table-striped">
36.         <thead>
37.             <tr>
38.                 <th scope="col">#</th>
39.                 <th scope="col">年份</th>
40.                 <th scope="col">月份</th>
41.                 <th scope="col">金额</th>
42.                 <th scope="col">备注</th>
43.             </tr>
44.         </thead>
45.         <tbody>
46.             <tr>
47.                 <th scope="row">1</th>
48.                 <td>2019</td>
49.                 <td>01</td>
50.                 <td>100.00</td>
51.                 <td>网络中心勤工助学</td>
52.             </tr>
53.             <tr>
54.                 <th scope="row">2</th>
55.                 <td>2020</td>
56.                 <td>05</td>
57.                 <td>210.00</td>
58.                 <td>资助中心小程序开发维护</td>
59.             </tr>
60.             <tr>
61.                 <th scope="row">3</th>
62.                 <td>2020</td>
63.                 <td>10</td>
64.                 <td>5000.00</td>
65.                 <td>校一等奖学金</td>
66.             </tr>
67.         </tbody>
68.     </table>
69. </div>
70. <script src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.
    js"
71.     integrity="sha384-
    DfXdz2htPH0lsSSs5nCTpuj/zy4C+0GpamoFVy38MVBnE+IbbVYUew+0rCXaRkfj"
72.     crossorigin="anonymous"></script>

```

```

73. <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.m
    in.js"
74.         integrity="sha384-
    Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
75.         crossorigin="anonymous"></script>
76. <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.0/dist/js/bootstrap.
    min.js"
77.         integrity="sha384-
    OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JkI"
78.         crossorigin="anonymous"></script>
79. </body>
80. </html>

```

资助金查询效果图如下所示：

#	年份	月份	金额	备注
1	2019	01	100.00	网络中心勤工助学
2	2020	05	210.00	资助中心小程序开发维护
3	2020	10	5000.00	校一等奖学金

图表 1-5 资助金查询

1.3 其他展示

1.3.1 JavaScript 验证登录

在登录页面验证输入的用户名和密码是否合法，使用正则表达对输入字段进行验证，相关正则验证 valid.js 代码如下：

```

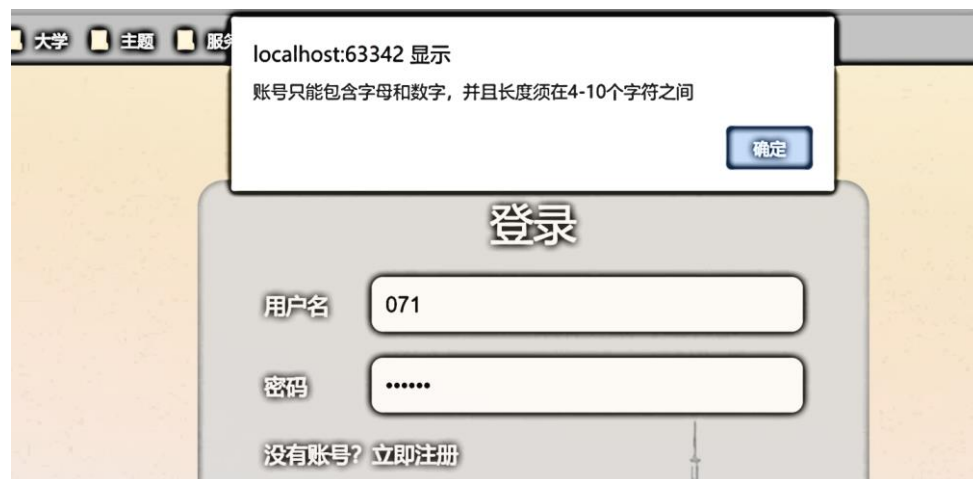
1. function validUsername(username) {
2.     let pattern = /^[A-Za-z0-9]{4,10}$/;
3.     return pattern.test(username);
4. }
5.
6. function validPassword(password) {
7.     // 包含字母和数字，可以使用特殊字符，长度在 6-16 之间
8.     let pattern = /^[A-Za-z0-9.@!%&]{6,16}$/;
9.     return pattern.test(password);
10. }
11.
12. function validPwdAndPwd2(pwd, pwd2) {
13.     // 判断密码 1 与密码 2 是否相同
14.     return pwd === pwd2;
15. }

```


又由于表单特性，提交会重新刷新页面，所以添加一个 Listener 来监听表单 submit 事件，当验证通过后会提交数据，否则阻止表单的提交事件，login.js 代码如下：

```
1. let form = document.getElementsByTagName('form')[0];
2. form.addEventListener("submit", function (event) {
3.     let username = event.target.username.value.trim();
4.     let password = event.target.password.value;
5.     let flag = false
6.     if (!validUsername(username)) {
7.         alert('账号只能包含字母和数字，并且长度须在 4-10 个字符之间')
8.         flag = true
9.     } else if (!validPassword(password)) {
10.        alert('密码长度需要在 6-16 个字符之间');
11.        flag = true
12.    }
13.    if (flag) {
14.        // 校验失败阻止表单提交事件，以取消数据传送
15.        event.preventDefault();
16.    } else {
17.        alert("登录成功")
18.    }
19. }, false);
```

验证效果图如下所示：



图表 1-6 用户名错误



图表 1-7 密码验证失败



图表 1-8 登录验证通过

1.3.2 JavaScript 实现轮播图

轮播图 banner.js 代码如下所示：

```

1. let images = document.getElementsByClassName("img")
2. let cur = 0;
3. function removeActivateClass(i) {
4.     // 移除激活状态
5.     let dotsItems = document.getElementsByClassName('dots-item');
6.     let flag = dotsItems[i].classList.contains('activate')
7.     if (flag) {
8.         dotsItems[i].classList.remove("activate")
9.         dotsItems[i].classList.add("deactivate")
10.    }
11. }
12. function addActivateClass(i) {
13.     // 添加激活状态
14.     let dotsItems = document.getElementsByClassName('dots-item');

```

```

15.     let flag = dotsItems[i].classList.contains('deactivate')
16.     if (flag) {
17.         dotsItems[i].classList.remove("deactivate")
18.         dotsItems[i].classList.add("activate")
19.     }
20. }
21. function setInter() {
22.     return setInterval(function () {
23.         removeActivateClass(cur)
24.         images[cur].hidden = true;
25.         cur = (cur + 1) % images.length
26.         images[cur].hidden = false;
27.         addActivateClass(cur)
28.     }, 3000)
29. }
30. let bannerInter = setInter()
31. function lastImage() {
32.     clearInterval(bannerInter);
33.     removeActivateClass(cur)
34.     images[cur].hidden = true;
35.     cur = cur === 0 ? images.length - 1 : (cur - 1) % images.length;
36.     images[cur].hidden = false;
37.     addActivateClass(cur)
38.     bannerInter = setInter()
39. }
40. function nextImage() {
41.     clearInterval(bannerInter);
42.     removeActivateClass(cur)
43.     images[cur].hidden = true;
44.     cur = (cur + 1) % images.length;
45.     images[cur].hidden = false;
46.     addActivateClass(cur)
47.     bannerInter = setInter(cur)
48. }
49. function activate(i) {
50.     clearInterval(bannerInter);
51.     removeActivateClass(cur)
52.     images[cur].hidden = true;
53.     images[i].hidden = false;
54.     addActivateClass(i)
55.     cur = i
56.     bannerInter = setInter(cur)
57. }

```

轮播图效果：图片会 3 秒自动切换为下一张，点击左右会切换为相应的图片，点击右下角数字会直接跳转至相应的图片，效果图如下所示。



图表 1-9 轮播图

1.4 设计心得

在上一学期稍有接触过 HTML，这一次的 Web 课使我更加系统地学习了开发 Web 应用的常用技术，此次实验很好的巩固了通过 HTML 标记语言来搭建网站同时使用 CSS 进行美化的技术。客户端脚本语言 JavaScript 也做了一些简单的尝试。

实验过程中发现了在 CSS 美化这一部分仍然有很大欠缺。在学习借鉴其他互联网资源时，发现他们的 CSS 都做得十分精细，而我美工的能力还是有很大的欠缺，这一点在这次实验中也有所体现。之后的学习过程中，要增加操练，不断加强自己去运用的能力，更好的掌握 Web 开发的技术。能够搭建出更加美观的站点。

2 动态 Web 页面设计实验

2.1 实验说明

2.1.1 实验目的

- 1、熟悉 JSP 的开发工具，掌握服务器端 Web 程序的工作原理
- 2、熟悉 JSP 编译指令，动作标记
- 3、熟悉 JSP 的隐含对象，正确理解 request、session、application 三个对象的作用域
- 4、掌握编写 JavaBean 的方法，使用 JSP `<jsp:useBean>`、`<jsp:set Property>`、`<jsp:getProperty>` 3 个动作指令。
- 5、掌握 JSP 中表单和表单 Bean 的映射

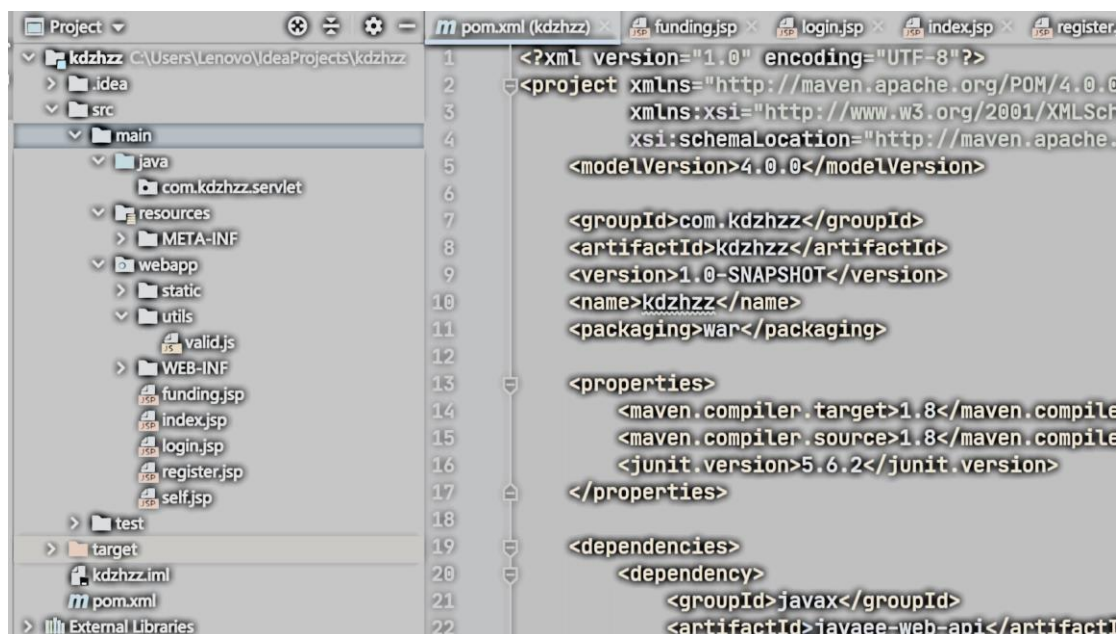
2.1.2 实验要求

- 1、实验之前认真查阅相关资料，准备好实验方案。
- 2、认真实验，对实验过程、结果进行分析，注意验证实验效果。
- 3、完成实验报告和实验成果。

2.2 实验内容

2.2.1 网站结构

修改实验一中的代码为 JSP 动态页面，修改后的目录结构如下图所示：



图表 2-1 目录结构

2.2.2 JavaBean 验证

新建 addregister.jsp 作为注册结果页面，通过 useBean 和 setProperty 建立 User 对象，并设置该对象的属性为提交上来的表单的属性，`property="**"`可以将

表单中同名称的属性值赋值给 JavaBean 对象中的同名属性，UserBean.java 代码如下：

```
1. package com.kdzhzz.entity;
2.
3. import java.util.ArrayList;
4. import java.util.List;
5.
6. public class UserBean {
7.     private String username;
8.     private String password;
9.     private List<String> res = new ArrayList<>();
10.
11.     public List<String> getRes() {
12.         return res;
13.     }
14.
15.     public void setRes(List<String> res) {
16.         this.res = res;
17.     }
18.
19.
20.     public String getPassword2() {
21.         return password2;
22.     }
23.
24.     public void setPassword2(String password2) {
25.         this.password2 = password2;
26.     }
27.
28.     private String password2;
29.
30.     public String getUsername() {
31.         return username;
32.     }
33.
34.     public void setUsername(String username) {
35.         this.username = username;
36.     }
37.
38.     public String getPassword() {
39.         return password;
40.     }
41.
42.     public void setPassword(String password) {
```

```

43.         this.password = password;
44.     }
45.
46.     public UserBean() {
47.     }
48.
49.     public UserBean(String username, String password) {
50.         this.username = username;
51.         this.password = password;
52.     }
53.
54.     public void isValid() {
55.         this.res.clear();
56.         boolean flag = true;
57.         if (!this.username.matches("[A-Za-z0-9]{4,10}$")) {
58.             flag = false;
59.             this.res.add("账号只能包含字母和数字，并且长度须在 4-10 个字符之间");
60.         }
61.         if (!this.password.matches("[A-Za-z0-9-9.@!%&]{6,16}$") || !this.password2.matches("[A-Za-z0-9-9.@!%&]{6,16}$")) {
62.             flag = false;
63.             this.res.add("密码长度需要在 6-16 个字符之间");
64.         }
65.         if (!this.password.equals(this.password2)) {
66.             flag = false;
67.             this.res.add("两次密码输入不一致");
68.         }
69.         if (flag) {
70.             this.res.add("注册成功");
71.         }
72.     }
73. }

```

注册验证 addregister.jsp 代码如下所示：

```

1. <%@ page contentType="text/html; charset=UTF-8" language="java" %>
2. <html>
3. <head>
4.     <title>Title</title>
5. </head>
6. <body>
7. <jsp:useBean id="newuser" scope="session" class="com.kdzhzz.entity.UserBean" />
8. <jsp:setProperty property="*" name="newuser"/>

```

```

9. 用户名: <jsp:getProperty name="newuser" property="username"/>
10. <br/>
11. 密码 1: <jsp:getProperty name="newuser" property="password"/>
12. <br/>
13. 密码 2: <jsp:getProperty name="newuser" property="password2"/>
14. <br/>
15. <%
16.     newuser.isValid();
17. %>
18. 注册状态: <jsp:getProperty name="newuser" property="res"/>
19. </body>
20. </html>

```

注册结果效果图如下所示:

```

用户名: 07172757
密码1: 17
密码2: 173510
注册状态: [密码长度需要在6-16个字符之间, 两次密码输入不一致]

```

图表 2-2 验证未通过

```

用户名: 07172757
密码1: 173510
密码2: 173510
注册状态: [注册成功]

```

图表 2-3 验证通过

2.2.3 Request 对象

在 index.js 页面使用 request 对象实现 login.jsp 登录数据提交参数获取, 其中使用 JSP 表达式来判断是否真正登录, 代码如下所示:

```

1. <div class="navbar-tool">
2.     <a href="index.jsp">首页</a>
3.     <a href="self.jsp">个人信息</a>
4.     <a href="funding.jsp">奖助金查询</a>
5. <%
6.     String username = request.getParameter("username");
7. %>
8. <a href="<%=username == null ? "login.jsp" : "#"%>">
9.     <%=username == null ? "登录" : username%>
10. </a>

```


11. </div>

效果图如下所示：



图表 2-4 登录前

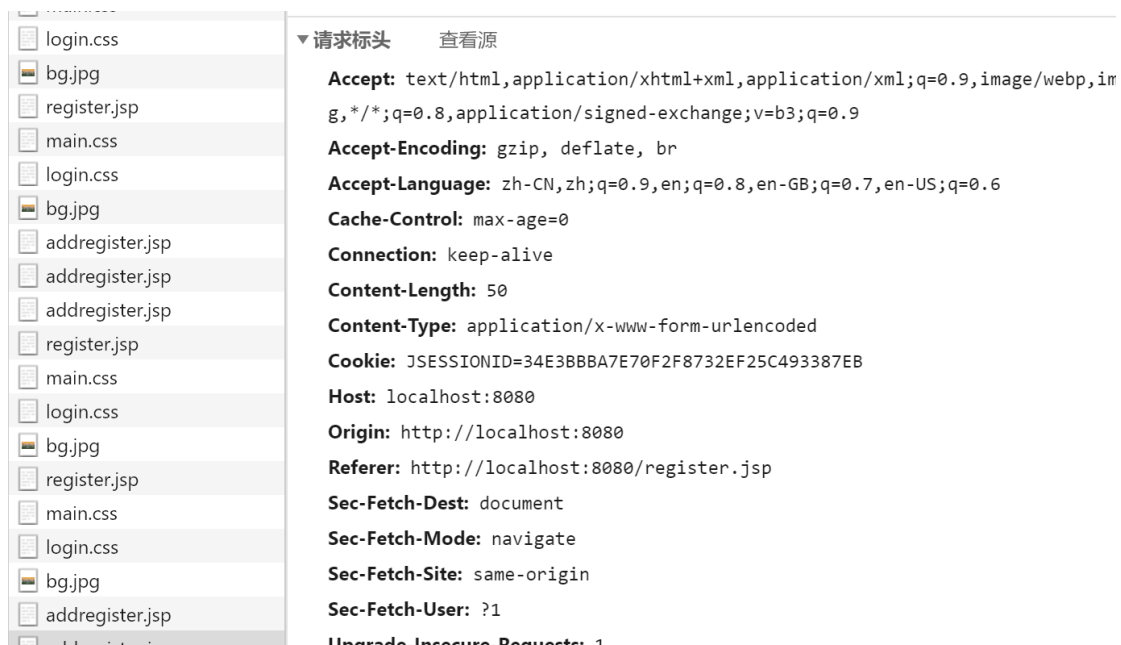


图表 2-5 登陆后

修改 addregister.jsp 代码使用 request 获取头信息，代码如下：

```
1. <br/>
2. <%
3.     String contentType = request.getHeader("content-type");
4.     String host = request.getHeader("host");
5. %>
6. contentType: <%=contentType%>
7. <br/>
8. host: <%=host%>
```

效果图如下所示：



图表 2-6 浏览器开发工具结果

```
contentType: application/x-www-form-urlencoded
host: localhost:8080
```

图表 2-7 实验获取结果

2.2.4 Response 对象

同样在 addregister.jsp 中实验，增加设置相应头字段 username 为注册的用户名，代码如下：

```
1. <%
2.     String username = request.getParameter("username");
3.     response.setHeader("username", username);
4. %>
```

效果图如下所示：

▼ 响应标头 查看源

```
Connection: keep-alive
Content-Length: 390
Content-Type: text/html; charset=UTF-8
Date: Tue, 03 Nov 2020 01:25:54 GMT
Keep-Alive: timeout=20
Set-Cookie: JSESSIONID=23A6F43EFB383AC079BA767D3AA21FA5; Path=/; HttpOnly
```

图表 2-8 添加 username 前

▼ 响应标头 查看源

```
Connection: keep-alive
Content-Length: 309
Content-Type: text/html; charset=UTF-8
Date: Tue, 03 Nov 2020 01:29:30 GMT
Keep-Alive: timeout=20
username: 07172757
```

图表 2-9 添加 username 后

2.2.5 Session 对象

登录 session

新建 addlogin.jsp 代码，用户将登录成功信息保存在 session 中，代码如下：

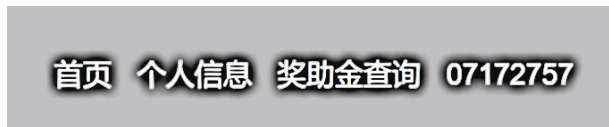
```
1. <%@ page contentType="text/html; charset=UTF-8" language="java" %>
2. <html>
3. <head>
4.     <title>登录验证</title>
5. </head>
6. <body>
```

```

7.  <%
8.      String username = request.getParameter("username");
9.      session.setAttribute("username",username);
10. %>
11. </body>
12. </html>

```

修改 index.jsp 将 request 获取 username 参数改为 session 获取，效果图如下所示：



图表 2-10 获取用户信息

注销 session

新建 logout.jsp 文件，设置 session 失效，代码如下：

```

1.  <%@ page contentType="text/html; charset=UTF-8" language="java" %>
2.  <html>
3.  <head>
4.      <title>注销 session</title>
5.  </head>
6.  <body>
7.  <%
8.      session.invalidate();
9.  %>
10. </body>
11. </html>

```

重新运行程序，先登录，后访问 logout.jsp 结果如下图所示，用户信息已被清除：



图表 2-11 注销登录

2.2.6 客户端跳转与服务器端跳转

客户端跳转

修改 addlogin.jsp 文件，添加客户端跳转代码，如下所示：

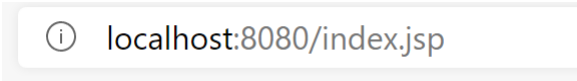
```

1.  <%
2.      String username = request.getParameter("username");
3.      session.setMaxInactiveInterval(30 * 60);
4.      session.setAttribute("username", username);
5.      response.sendRedirect("index.jsp"); // 客户端跳转

```

6. %>

访问 login.jsp 进行登录，观察浏览器 url 地址，如下所示，地址变为 index.jsp:



图表 2-12 客户端跳转结果

修改 addlogin.jsp 文件，添加服务端跳转代码，如下所示:

```
1. <%  
2.     String username = request.getParameter("username");  
3.     session.setMaxInactiveInterval(30 * 60);  
4.     session.setAttribute("username", username);  
5.     request.getRequestDispatcher("index.jsp").forward(request, response);  
6. %>
```

访问 login.jsp 进行登录，发现登录成功页不再是 index.jsp，说明服务端跳转成功，如下图所示:



图表 2-13 服务端跳转结果

2.3 设计心得

这次实验很好的练习了 **JavaBean** 的使用，通过构造 **Bean** 类，可以方便的将表单信息传输到所需要的对象并完成设置。同时，熟悉了客户端与服务器端的跳转，通过一些简单的登录注销，进一步熟悉了动态网页的搭建造。

这一次实验，在登录验证部分选择了全部放行，仅为了学习跳转，在接下来的数据库操作实验中期待进一步通过访问数据库，更好的去进行登陆判断。

3 Web 数据库开发实验

3.1 实验说明

3.1.1 实验目的

- 1、掌握 Servlet 的开发、配置
- 2、掌握 Filter 的开发与配置
- 3、熟悉 JDBC 以及 DAO 的概念及工作原理
- 4、能够熟练运用 Servlet+DAO 模式对数据库进行访问，实现数据查询、添加、修改等常用操作
- 5、能够熟练运用 MVC 模式开发 Web 应用程序，实现数据查询、添加、修改等常用操作。

3.1.2 实验要求

- 1、实验之前认真查阅相关资料，准备好实验方案。
- 2、认真实验，对实验过程、结果进行分析，注意验证实验效果。
- 3、完成实验报告和实验成果

3.1.3 实验内容

- 1、请设计一种过滤器实现权限控制机制。如果用户进入 Web 应用没有登录时，要求用户必须进行登录页面。
- 2、使用 JDBC 技术进行数据库的连接与访问。
- 3、采用 DAO 设计模式开发。使用自己熟悉的网络数据库，设计一个小的系统（可以在之间的页面上实现），要求如下：
 - a) 至少包括 2 张数据表（字段数不少于 4）。
 - b) 实现对 2 张表格的增、删、改、查操作。
 - c) 具有不同类型用户的登录控制页面。
 - d) 要求每一项操作对应于一张页面。

3.2 详细设计与编码

3.2.1 设计数据库表格

实验设置两个数据表格，user_info 用户信息表和 funding_list 资助金信息表，如下图所示：

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	
username	varchar	16	0	<input type="checkbox"/>	<input type="checkbox"/>		
name	varchar	32	0	<input type="checkbox"/>	<input type="checkbox"/>		
college	varchar	16	0	<input type="checkbox"/>	<input type="checkbox"/>		
rank	varchar	32	0	<input type="checkbox"/>	<input type="checkbox"/>		
profession	varchar	32	0	<input type="checkbox"/>	<input type="checkbox"/>		
clazz	varchar	32	0	<input type="checkbox"/>	<input type="checkbox"/>		
password	varchar	32	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

图表 3-1 用户信息表

保存

添加字段


插入字段

删除字段

主键

上移

下移

字段	索引	外键	触发器	选项	注释	SQL 预览					
名					类型	长度	小数点	不是 null	虚拟	键	注释
id					int	11	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	
username					varchar	16	0	<input type="checkbox"/>	<input type="checkbox"/>		
year					varchar	4	0	<input type="checkbox"/>	<input type="checkbox"/>		
month					varchar	4	0	<input type="checkbox"/>	<input type="checkbox"/>		
amount					double	8	2	<input type="checkbox"/>	<input type="checkbox"/>		
remark					varchar	32	0	<input type="checkbox"/>	<input type="checkbox"/>		

图表 3-2 资助金信息

3.2.2 连接数据库

首先在 pom.xml 中引入 MySQL 连接的依赖包，代码如下所示：

```

1. <dependency>
2.     <groupId>mysql</groupId>
3.     <artifactId>mysql-connector-java</artifactId>
4.     <version>5.0.8</version>
5. </dependency>

```

数据库 mysql 连接 DatabaseConnection.java 代码如下所示：

```

1. package com.kdzhzz.sql;
2.
3. import java.sql.*;
4.
5. public class DatabaseConnection {
6.     public static void main(String[] args) {
7.         try {
8.             Class.forName("com.mysql.jdbc.Driver");
9.         } catch (ClassNotFoundException e) {
10.            e.printStackTrace();
11.        }
12.        String url = "jdbc:mysql://121.41.227.161:3306/cumt_funding";
13.        String username = "cumt_xxxx";
14.        String password = "xxxxxxx";
15.        Connection connection = null;
16.        try {
17.            connection = DriverManager.getConnection(url, username, password
18.        );
19.        } catch (SQLException throwables) {
20.            throwables.printStackTrace();
21.        }
22.        System.out.println(connection);
23.        try {
24.            assert connection != null;
25.            connection.close();

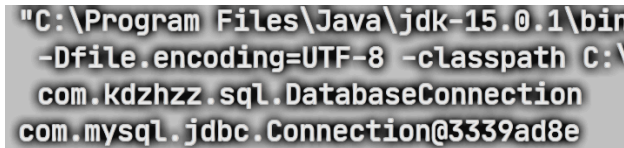
```

```

25.         } catch (SQLException throwables) {
26.             throwables.printStackTrace();
27.         }
28.     }
29. }

```

连接结果：



```

"C:\Program Files\Java\jdk-15.0.1\bin
-Dfile.encoding=UTF-8 -classpath C:\
com.kdzhzz.sql.DatabaseConnection
com.mysql.jdbc.Connection@3339ad8e

```

图表 3-3 数据库连接结果

3.2.3 Filter 登录权限控制机制

新建 LoginFilter.java 对主页、奖助金信息页与个人信息页面进行控制。通过在 filter 中查询 session 中是否有个人登录成功信息来进行页面的客户端重定向，代码如下所示（使用注解方式对路由进行控制）：

```

1. package com.kdzhzz.filter;
2.
3. import javax.servlet.*;
4. import javax.servlet.annotation.WebFilter;
5. import javax.servlet.http.HttpServletRequest;
6. import javax.servlet.http.HttpServletResponse;
7. import javax.servlet.http.HttpSession;
8. import java.io.IOException;
9.
10. @WebFilter({" /index.jsp", "/funding.jsp", "/self.jsp"})
11. public class LoginFilter implements Filter {
12.     @Override
13.     public void doFilter(ServletRequest servletRequest, ServletResponse serv
        letResponse, FilterChain filterChain) throws IOException, ServletException {
14.         HttpServletRequest httpRequest = (HttpServletRequest) servlet
            Request;
15.         HttpServletResponse httpResponse = (HttpServletResponse) serv
            letResponse;
16.         HttpSession session = httpRequest.getSession();
17.         String username = (String) session.getAttribute("username");
18.         if (username == null) {
19.             // 未登录，跳转至登录页面
20.             httpResponse.sendRedirect("login.jsp");
21.         }
22.         filterChain.doFilter(httpServletRequest, httpResponse);
23.     }

```

```
24. }
```

3.2.4 数据库连接与关闭公共方法

由于数据库的操作都需要进行连接与关闭，所以抽象出一个公共方法 DatabaseConnection.java 代码如下所示：

```
1. package com.kdzhzz.database;
2.
3. import java.sql.*;
4.
5. public class DatabaseConnection {
6.     private static String driver = "com.mysql.cj.jdbc.Driver";
7.     private static String url = "jdbc:mysql://xxxxxx:3306/cumt_funding";
8.     private static String username = "xxxx";
9.     private static String password = "xxxxx";
10.
11.     static {
12.         try {
13.             Class.forName(driver);
14.         } catch (ClassNotFoundException e) {
15.             e.printStackTrace();
16.         }
17.     }
18.
19.     public static Connection getConnection() throws SQLException {
20.         return DriverManager.getConnection(url, username, password);
21.     }
22.
23.     public static void closeAll(Connection conn, Statement stmt, ResultSet r
24. s) throws SQLException {
25.         if (rs != null) {
26.             rs.close();
27.         }
28.         if (stmt != null) {
29.             stmt.close();
30.         }
31.         if (conn != null) {
32.             conn.close();
33.         }
34. }
```


3.2.5 用户表增查改

用户 UserDao.java 代码如下所示，定义了用户的注册，认证与修改接口：

```
1. package com.kdzhzz.dao.user;
2.
3. import com.kdzhzz.entity.UserInfo;
4.
5. public interface UserDao {
6.     void createUser(String username, String password) throws Exception;
7.     UserInfo authUser(String username, String password) throws Exception;
8.     UserInfo putUser(Integer id, String college, String clazz) throws Exception;
9. }
```

UserDaoImpl.java 代码如下，实现了 UserDao 接口的方法。

```
1. public class UserDaoImpl implements UserDao {
2.     public UserInfo getUserInfo(ResultSet resultSet) throws SQLException {
3.         UserInfo userInfo = null;
4.         if (resultSet.next()) {
5.             userInfo = new UserInfo(
6.                 resultSet.getInt("id"),
7.                 resultSet.getString("name"),
8.                 resultSet.getString("username"),
9.                 resultSet.getString("college"),
10.                resultSet.getString("rank"),
11.                resultSet.getString("profession"),
12.                resultSet.getString("clazz"),
13.                resultSet.getString("typz")
14.            );
15.        }
16.        return userInfo;
17.    }
18.
19.    @Override
20.    public void createUser(String username, String password) throws Exception {
21.        Connection connection = DatabaseConnection.getConnection();
22.        String sql = "INSERT INTO user_info (username,password,typz) VALUES (?, ?, ?)";
23.        PreparedStatement pstmt = connection.prepareStatement(sql);
24.        pstmt.setString(1, username);
25.        pstmt.setString(2, password);
```

```

26.         pstmt.setString(3, "student");
27.         Pstmt.executeUpdate();
28.         DatabaseConnection.closeAll(connection, pstmt, null);
29.     }
30.
31.     @Override
32.     public UserInfo authUser(String username, String password) throws Except
        ion {
33.         Connection connection = DatabaseConnection.getConnection();
34.         String sql = "select * from user_info where username=? and password=
            ?";
35.         PreparedStatement pstmt = connection.prepareStatement(sql);
36.         pstmt.setString(1, username);
37.         pstmt.setString(2, password);
38.         UserInfo userInfo = null;
39.         ResultSet resultSet = pstmt.executeQuery();
40.         userInfo = getUserInfo(resultSet);
41.         DatabaseConnection.closeAll(connection, pstmt, resultSet);
42.         return userInfo;
43.     }
44.
45.     @Override
46.     public UserInfo putUser(Integer id, String college, String clazz) throws
        Exception {
47.         Connection connection = DatabaseConnection.getConnection();
48.         String sql = "update user_info set college = ?, clazz = ? where id=?
            ";
49.         PreparedStatement pstmt = connection.prepareStatement(sql);
50.         pstmt.setString(1, college);
51.         pstmt.setString(2, clazz);
52.         pstmt.setInt(3, id);
53.         pstmt.executeUpdate();
54.
55.         sql = "select * from user_info where id=?";
56.         pstmt = connection.prepareStatement(sql);
57.         pstmt.setInt(1, id);
58.         ResultSet resultSet = pstmt.executeQuery();
59.
60.         UserInfo userInfo = getUserInfo(resultSet);
61.         DatabaseConnection.closeAll(connection, pstmt, resultSet);
62.         return userInfo;
63.     }
64. }

```

用户认证 servlet 主要代码如下：

```
1. protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
2.     String username = req.getParameter("username");
3.     String password = req.getParameter("password");
4.     UserDaoImpl userDaoImpl = new UserDaoImpl();
5.     UserInfo userInfo = null;
6.     try {
7.         userInfo = userDaoImpl.authUser(username, password);
8.     } catch (Exception e) {
9.         e.printStackTrace();
10.    }
11.
12.    HttpSession session = req.getSession();
13.    assert userInfo != null;
14.    session.setAttribute("userInfo", userInfo);
15.    session.setAttribute("username", userInfo.getUsername());
16.    resp.sendRedirect("/index.jsp");
17. }
```

用户注册 servlet 主要代码如下，默认类别为学生：

```
1. protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
2.     String username = req.getParameter("username");
3.     String password = req.getParameter("password");
4.     UserDaoImpl userDaoImpl = new UserDaoImpl();
5.     try {
6.         userDaoImpl.createUser(username, password);
7.     } catch (Exception e) {
8.         e.printStackTrace();
9.     }
10.    resp.sendRedirect("/login.jsp");
11. }
```

用户信息修改 servlet 主要代码如下：

```
1. protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
2.     Integer id = Integer.valueOf(req.getParameter("id"));
3.     String college = req.getParameter("college");
4.     String clazz = req.getParameter("clazz");
5.
6.     UserDaoImpl userDaoImpl = new UserDaoImpl();
```

```

7.     try {
8.         UserInfo userInfo = userInfoDaoImpl.putUser(id, college, clazz);
9.         HttpSession session = req.getSession();
10.        session.setAttribute("userInfo", userInfo);
11.    } catch (Exception e) {
12.        e.printStackTrace();
13.    }
14.    resp.sendRedirect("/self.jsp");
15. }

```

用户登录成功获取用户信息，如下图所示：

姓名：李治远	学号：07172757
学院：计算机科学与技术学院	年级：2017
专业：计算机科学与技术	班级：计科17-1班

图表 3-4 用户信息查询

用户注册成功结果如下图所示（123456 为新注册用户）：

开始事务	文本	筛选	排序	导入	导出			
id	username	name	college	rank	profession	clazz	password	typz
1	07172757	李治远	计算机科学与技术学院	2017	计算机科学与技术	计科17-1班	173510	admin
2	123456	(Null)	(Null)	(Null)	(Null)	(Null)	173510	student

图表 3-5 用户信息创建

用户信息修改如下图所示：

学院：	<input type="text" value="xxxx学院"/>
班级：	<input type="text" value="新的班级"/>
<input type="button" value="更新"/>	

图表 3-6 用户信息修改表单

姓名：李治远	学号：07172757
学院：xxxx学院	年级：2017
专业：计算机科学与技术	班级：新的班级

图表 3-7 用户信息修改结果

3.2.6 奖励金信息查增删

奖励学金 FundingDao.java 代码如下所示，定义了添加、根据学号查询、所有查询与删除接口。

```

1. package com.kdzhzz.dao.funding;
2. import com.kdzhzz.entity.Funding;
3.
4. import java.util.List;
5.
6. public interface FundingDao {
7.     void createFunding(String username, String year, String month, String amount, String remark) throws Exception;
8.     List<Funding> findFunding(String username) throws Exception;
9.     List<Funding> findAll() throws Exception;
10.    void deleteFunding(Integer id) throws Exception;
11. }

```

奖助学金 FundingDaoImpl.java 代码如下所示，实现了添加、根据学号查询、所有查询与删除方法。

```

1. public class FundingDaoImpl implements FundingDao {
2.     public List<Funding> getFunding(ResultSet resultSet) throws SQLException
3.     {
4.         List<Funding> funding = new ArrayList<>();
5.         while (resultSet.next()) {
6.             funding.add(
7.                 new Funding(
8.                     resultSet.getInt("id"),
9.                     resultSet.getString("username"),
10.                    resultSet.getString("year"),
11.                    resultSet.getString("month"),
12.                    resultSet.getString("amount"),
13.                    resultSet.getString("remark")
14.                );
15.            }
16.        return funding;
17.    }
18.
19.    @Override
20.    public void createFunding(String username, String year, String month, String amount, String remark) throws Exception {
21.        Connection connection = DatabaseConnection.getConnection();
22.        String sql = "INSERT INTO funding_list (username,year, month,amount, remark) VALUES (?,?,,?,?);";
23.        PreparedStatement pstmt = connection.prepareStatement(sql);
24.        pstmt.setString(1, username);
25.        pstmt.setString(2, year);
26.        pstmt.setString(3, month);

```

```

27.         pstmt.setString(4, amount);
28.         pstmt.setString(5, remark);
29.         pstmt.executeUpdate();
30.         DatabaseConnection.closeAll(connection, pstmt, null);
31.     }
32.
33.     @Override
34.     public List<Funding> findFunding(String username) throws Exception {
35.         Connection connection = DatabaseConnection.getConnection();
36.         String sql = "select * from funding_list where username=?";
37.         PreparedStatement pstmt = connection.prepareStatement(sql);
38.         pstmt.setString(1, username);
39.         List<Funding> funding = null;
40.         ResultSet resultSet = pstmt.executeQuery();
41.         funding = getFunding(resultSet);
42.         DatabaseConnection.closeAll(connection, pstmt, resultSet);
43.         return funding;
44.     }
45.
46.     @Override
47.     public List<Funding> findAll() throws Exception {
48.         Connection connection = DatabaseConnection.getConnection();
49.         String sql = "select * from funding_list";
50.         PreparedStatement pstmt = connection.prepareStatement(sql);
51.         List<Funding> funding = null;
52.         ResultSet resultSet = pstmt.executeQuery();
53.         funding = getFunding(resultSet);
54.         DatabaseConnection.closeAll(connection, pstmt, resultSet);
55.         return funding;
56.     }
57.
58.     @Override
59.     public void deleteFunding(Integer id) throws Exception {
60.         Connection connection = DatabaseConnection.getConnection();
61.         String sql = "delete from funding_list where id = ?";
62.         PreparedStatement pstmt = connection.prepareStatement(sql);
63.         pstmt.setInt(1, id);
64.         pstmt.executeUpdate();
65.         DatabaseConnection.closeAll(connection, pstmt, null);
66.     }
67. }

```

添加奖助金信息 servlet 主要代码为：

```

1. protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
2.     FundingDaoImpl fundingDaoImpl = new FundingDaoImpl();
3.     String username = req.getParameter("username");
4.     String year = req.getParameter("year");
5.     String month = req.getParameter("month");
6.     String amount = req.getParameter("amount");
7.     String remark = req.getParameter("remark");
8.     try {
9.         fundingDaoImpl.createFunding(username, year, month, amount, remark);
10.    } catch (Exception e) {
11.        e.printStackTrace();
12.    }
13.    resp.sendRedirect("/allfunding.jsp");
14. }

```

添加奖助金信息效果图：

添加奖助金

学号:

年:

月:

金额:

备注:

图表 3-8 添加奖助金

#	学号	年份	月份	金额	备注	操作
2	07172757	2019	01	100.0	网络中心勤工助学	删除 编辑
3	07172757	2020	05	210.0	资助中心小程序开发维护	删除 编辑
4	123456	2020	09	5000.0	励志奖学金	删除 编辑

图表 3-9 添加奖助金结果

删除奖助金信息 servlet 主要代码为：

```

1. @WebServlet("/funding/delete")
2. public class FundingDeleteServlet extends HttpServlet {
3.     @Override
4.     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

```

```

5.         Integer id = Integer.valueOf(req.getParameter("id"));
6.         FundingDaoImpl fundingDaoImpl = new FundingDaoImpl();
7.         try {
8.             fundingDaoImpl.deleteFunding(id);
9.         } catch (Exception e) {
10.            e.printStackTrace();
11.        }
12.        resp.sendRedirect("/funding/list");
13.    }
14. }

```

删除奖助金信息效果图如下：

#	学号	年份	月份	金额	备注	操作
2	07172757	2019	01	100.0	网络中心勤工助学	删除 编辑
3	07172757	2020	05	210.0	资助中心小程序开发维护	删除 编辑

图表 3-10 删除奖助金信息

3.3 其他展示

3.3.1 角色访问控制

编写 FundingFilter.java，对用户访问进行控制，代码如下：

```

1. public class FundingFilter implements Filter {
2.     @Override
3.     public void doFilter(ServletRequest servletRequest, ServletResponse serv
        letResponse, FilterChain filterChain) throws IOException, ServletException {
4.         HttpServletRequest httpServletRequest = (HttpServletRequest) servlet
        Request;
5.         HttpSession session = httpServletRequest.getSession();
6.         UserInfo userInfo = (UserInfo) session.getAttribute("userInfo");
7.         String typz = userInfo.getTypz();
8.
9.         HttpServletResponse httpServletResponse = (HttpServletResponse) serv
        letResponse;
10.        if (!typz.equals("admin")) {
11.            httpServletResponse.sendRedirect("error.jsp");
12.        }
13.        filterChain.doFilter(httpServletRequest, httpServletResponse);
14.    }
15. }

```


配置 web.xml 文件，对添加奖助金信息和删除奖助金设置权限，如下所示：

```
1. <filter>
2.     <filter-name>fundingFilter</filter-name>
3.     <filter-class>com.kdzhzz.filter.FundingFilter</filter-class>
4. </filter>
5. <filter-mapping>
6.     <filter-name>fundingFilter</filter-name>
7.     <url-pattern>/funding/list</url-pattern>
8.     <url-pattern>/funding/add</url-pattern>
9. </filter-mapping>
```

效果图，登录新创建的 123456 用户，由于其类型为 student，所以提示无权访问，如下图所示：



图表 3-11 无权访问

3.4 设计心得

这次实验花费了比较多的精力，部分课堂上没有及时掌握的内容，通过课后再一次的消化吸收理解回放内容，在实践中进一步练习了 JDBC 的访问和使用，同时对于 Servlet 的运用和 Filter 的运用也有了很好地实践和体会。

过滤器确实很大程度上节省了编写重复代码，DAO 模式的引用的确是很好地将 JSP 页面中的 Java 语句与 HTML 语句相分离，整个设计都十分精妙。

在实验中，考虑到信息的安全性，仅对管理员进行了授权删除操作，相信在能够访问数据库的基础上，实现其他功能的具体语句并不具有难度。