福建师范大学协和学院

本科实验报告

课程名称： 人机交互

系 别： 信息技术系

年 级： 2018级

专 业：\_\_\_\_计科专升本

学 号： 124622018006

姓 名： 陈锐旺

指导老师： 李 敏

提交时间： 2019.6.12

**2018-2019第 2学期**

《人机交互》实验报告填写要求

# 一、基本说明

本课程共需完成实验项目八个。每一次实验必须提交实验报告、实验源码、数据文件，实验报告手册期末上交，必须同时交电子版和纸质版。

# 二、实验报告填写要求

1、实验部分包括实验名称、实验目的、实验工具、实验要求、实验结论等几个部分；

2、实验代码较多，代码不能截图，需要复制粘贴；

3、实验结果部分需要截图；

4、思考题部分必须手写；

5、本课程实验暂不实行分组，原则上个人独立完成，如遇特殊原因无法独立完成时，向指导老师提出，由指导老师安排分组；

6、因本课程的实验多为综合性实验，需要根据老师的要求进行粘贴代码或截图；

7、每份报告需填写实验时间。

# 三、其他要求

1、实验报告手写部分，要求字迹工整，绘图需整齐、清楚。

2、实验报告封面按学院要求打印。

3、对于抄袭实验报告和编撰数据的行为，一经发现，以零分处理，并根据相关条例给予处分。

4、每次实验，必须新的一页开始，双面打印。

5、思考题必须手写。

# 四、成绩评定

1、实验报告均由指导老师评定成绩，成绩分为A+、A-、B+、B-、C+、C-六个等级。

2、实验分与平时成绩占期末总评的30%。

## 《人机交互》实验项目列表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 实验项目名称 | 学时 | 实验  时间 | 实验  地点 | 成绩 | 指导老师 |
| 实验一 | 交互系统—页面设计与实现 | 4 | 1-2周 | 机房 |  |  |
| 实验二 | 交互系统—数据添加 | 4 | 3-4周 | 机房 |  |  |
| 实验三 | 交互系统—数据查询与删除 | 4 | 5-6周 | 机房 |  |  |
| 实验四 | 交互系统—数据修改 | 4 | 7-8周 | 机房 |  |  |
| 实验五 | 交互系统—AJAX交互 | 4 | 9-10周 | 机房 |  |  |
| 实验六 | 交互系统—Ueditor图片上传 | 4 | 11-12周 | 机房 |  |  |
| 实验七 | 交互系统—VUE.js查询新闻-PC | 4 | 13-14周 | 机房 |  |  |
| 实验八 | 交互系统—百度Echart（PC+APP） | 4 | 15-16周 | 机房 |  |  |

福建师范大学协和学院《人机交互》实验报告

学号：\_\_\_\_\_\_\_\_ 姓名：\_\_\_\_\_\_\_\_\_\_ 实验日期：\_2019.3.7\_\_ 成绩：\_\_\_\_\_\_\_

实验一 交互系统—页面设计与实现

# 一、实验目的

掌握HTML中的图像img、列表ul和li、换行p、div布局等基础标签的编程；

掌握H5高级部分的相关标签：header、section、footer、article等标签的使用；

掌握CSS基础id选择器、类选择器、分类选择器、元素选择器等用法；

掌握CSS中基本样式，外边距、内边距、边框、字体大小等用法；

掌握CSS3高级部分transform、border-radius的相关的用法。

# 二、实验环境（主要仪器设备和软件）

1、操作系统Windows 7

2、HBuilder.

3、Google Chrome

# 三、实验内容

题目一：设计一个PC端的web页面，可以是公司首页或者产品宣传页面等；

题目二：设计一个APP页面，页面上至少包含一副图片；

题目三：设计一个APP页面，要求使用MUI框架；

# 四、代码及运行结果

## 4.1 题目1

### 4.1.1 页面运行效果截图





### 4.1.2 HTML代码

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title></title>

<link rel="stylesheet" type="text/css" href="css/compjianjie.css"/>

</head>

<body>

<div id="box">

<div id="top">

<div id="top1">

<img src="img/logo.jpg" width="750px" height="150px" alt="" >

</div>

<div id="top2">

设为主页| 加入收藏

</div>

<div id="top3">

<ul>

<li>首页</li>

<li>公司简介</li>

<li>新闻中心</li>

<li>工程展示</li>

<li>企业管理</li>

<li>业务范围</li>

<li>在线留言</li>

<li>关于我们</li>

</ul>

</div>

<div id="top4">

<img src="img/heifu.jpg" width="910px" height="200px" >

</div>

</div>

<div id="main">

<h2>&gt;&gt;公司简介</h2>

<div id="imgId">

<img src="img/compic.jpg">

</div>

<p>保盟网络科技有限公司办公室地址位于中国东亚文化之都、海上丝绸之路起点---泉州，泉州 泉州市丰泽区城东街道学府上城8栋502室，于2017年01月06日在泉州市工商行政管理局丰泽分局注册成立，注册资本为1000万人民币，在公司发展壮大的2年里，我们始终为客户提供好的产品和技术支持、健全的售后服务，我公司主要经营通过实体店铺及互联网销售：计算机软硬件及辅助设备、机械设备、五金交电、电子产品、文化用品、体育用品、照相器材、化妆品、化工产品（不含危险化学品）、日用品、纺织品、服装、家具、金银珠宝首饰、一类医疗器械、初级农产品、预包装食品、乳制品（含婴幼儿配方乳粉）、音像制品、图书、报刊；房地产中介服务；设计、...，我们有好的产品和专业的销售和技术团队，我公司属于泉州计算机网络设备厂黄页行业，如果您对我公司的产品服务有兴趣，期待您在线留言或者来电咨询</p>

</div>

<div id="left\_2">

<h2>&gt;&gt;联系我们</h2>

<p>电话：0591-85645215 0591-38322024</p>

<p>地址：福州市鼓楼区xxx路xx大厦</p>

</div>

<div id="right\_1">

<h2>更多&gt;&gt;</h2>

<dl>

<dt>业务正在不断扩大</dt>

<dd>2018.11-27</dd>

<dt>业务正在不断扩大</dt>

<dd>2018.11-27</dd>

<dt>业务正在不断扩大</dt>

<dd>2018.11-27</dd>

<dt>业务正在不断扩大</dt>

<dd>2018.11-27</dd>

<dt>业务正在不断扩大</dt>

<dd>2018.11-27</dd>

</dl>

</div>

<div id="right\_2">

<h2>更多&gt;&gt;</h2>

<ul>

<li><img src="img/2.jpg" class="img2" width="150px" height="100px"> </li>

<li><img src="img/3.jpg" class="img3" width="150px" height="100px"></li>

<li><img src="img/4.jpg" class="img4" width="150px" height="100px"></li>

</ul>

</div>

<div id="footer">

Copyright 版权所有 保盟网络科技有限公司

</div>

</div>

</div>

</body>

</html>

### 4.1.3 CSS代码

body{

*background*: azure;

}

#box{

*width*: 910px;

*margin*: auto;

*background-color*: beige;

}

#top1{

*float*: *left*;

}

#top2{

*list-style-type*: none;

*float*: *right*;

*margin*: 20px 20px 0 0;

*font-family*: "微软雅黑";

*font-size*: 12px;

*color*: #666666;

}

#top3{

*clear*: both;

*overflow*: hidden;

*background*: #F5F5DC;

*margin-top*: 10px;

*margin-bottom*: 5px;

*background*: url(../img/biankuang.jpg);

}

#top3 ul{

*margin*: 0;

*margin*: 0;

}

#top3 ul li {

*list-style-type*: none;

*float*: *left*;*height*: 25px;

*width*: 102px;

*text-align*: center;

*font-family*: 微软雅黑;

*font-size*: 16px;

*color*: black;

*margin-top*: 5px;

*margin-left*: 5px;

}

#top4{

*text-align*: center;

}

#main{

*overflow*: hidden;

}

#left{

*width*: 250px;

*float*: *left*;

*font-size*:12px ;

*font-family*: 微软雅黑;

}

#right{

*width*: 630px;

*height*: 410px;

*float*: *right*;

}

#left\_1 p{

*margin*: 0px;

*text-align*: 2em;

*float*: *right*;

}

#left\_2{

*font-size*: 12px;

*border*: 1px solid #ccc;

}

#left-2 p{

*padding-left*: 5px;

}

#left\_2 h2{

*background*: url(../img/biankuang.jpg);

}

#main{

*border*: 2px ;

}

#main h2{

*margin-top*: 30px;

*margin-left*: 0px;

*font-size*: 14px;

*padding-top*: 2px;

*padding-left*: 5px;

*height*: 25px;

*color*: #000000;

*background*: url(../img/biankuang.jpg);

}

#main #imgId{

*width*:32%;

*height*: 200px;

*float*: *left*;

*padding*: 30px;

}

#main img{

*width*:100%;

}

#main p{

*width*:60%;

*float*: *right*;

}

#left\_2{

*font-size*: 12px;

*font-family*: "微软雅黑";

*border*: 1px;

}

#right\_1{

*border*: 1px solid #CCCCCC;

}

#right\_2{

*border*: 1px solid #CCCCCC;

}

#right\_1 h2,#right\_2 h2{

*font-size*: 12px;

*height*: 25px;

*font-family*: "微软雅黑";

*line-height*: 25px;

*padding-right*: 5px;

*font-size*: 16px;

*background*: url(../img/biankuang.jpg);

}

#right\_2 h2 {

*background*: url(../img/biankuang.jpg);

}

#right\_1 dl{

*font-size*: 12px;

*color*: #666666;

*overflow*: hidden;

}

#right\_1 dl dt{

border-bottom-width: 1px;

*border-bottom-style*: dashed;

*border-bottom-color*: #CCCCCC;

}

#right\_1 dl dd{

*border-bottom-width*: 1px;

*border-bottom-style*: dashed;

*border-bottom-color*: #CCCCCC;

*margin*: 0;

}

#right\_1 dl dt{

*width*: 540px;

*height*: 20px;

*float*: *left*;

*margin-top*: 8px;

*margin-left*: 10px;

}

#right\_1 dl dd{

*width*: 70px;

*height*: 20px;

*float*: *left*;

*margin-top*: 8px;

*text-align*: center;

}

#right\_2 ul {

*list-style-type*: none;

*overflow*: hidden;

}

#box #right\_2 ul li {

*height*: 150px;

*width*: 170px;

*float*: *left*;

*margin*: 10px;

*text-align*: center;

*font-size*: 12px;

*color*: #666666;

*padding*: 10px;

}

#right\_2 ul li img{

magrgin:10 px;

*padding*: 10px;

}

#footer{

*font-family*: "微软雅黑";

*font-size*: 14px;

*color*: #000000;

*height*: 30px;

*line-height*: 30px;

*text-align*: center;

*margin-top*: 10px;

*border-top-width*: 2px;

*border-top-style*: solid;

*border-top-color*: #0C71C3;

}

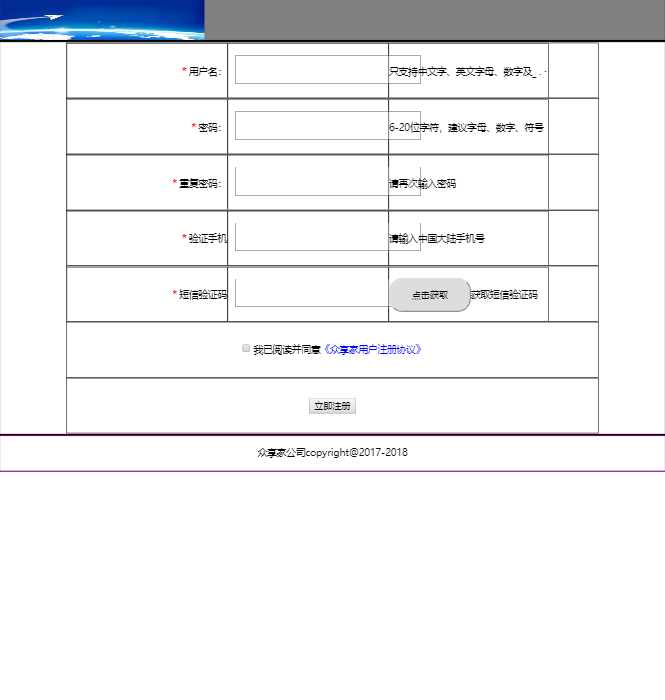
.img2{

*padding*: 30px;

}

## 4.2题目2

### 4.2.1 页面运行效果



### 4.2.2 HTML代码

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>注册页面</title>

<link rel="stylesheet" type="text/css" href="css/1.css"/>

</head>

<body>

<div id="main">

导航链接部分

</div>

<div id="nav">

<img src="img/2.jpg" width="300px" height="60px">

</div>

<div id="center">

<ul class="line">

<li class="lab"><b>\*</b>用户名：</li>

<li class="ipt"><input type="text" value="" /></li>

<li class="des">只支持中文字、英文字母、数字及\_ . ·</li>

</ul>

<ul class="line">

<li class="lab"><b>\*</b>密码：</li>

<li class="ipt"><input type="text" value="" /></li>

<li class="des">6-20位字符，建议字母、数字、符号混合使用</li>

</ul>

<ul class="line">

<li class="lab"><b>\*</b>重复密码：</li>

<li class="ipt"><input type="text" value="" /></li>

<li class="des">请再次输入密码</li>

</ul>

<ul class="line">

<li class="lab"><b>\*</b>验证手机</li>

<li class="ipt"><input type="text" value="" /></li>

<li class="des">请输入中国大陆手机号</li>

</ul>

<ul class="line">

<li class="lab"><b>\*</b>短信验证码</li>

<li class="ipt"><input type="text" value="" /></li>

<li class="des"><input type="button" value="点击获取" />获取短信验证码</li>

</ul>

<ul class="line agree">

<input type="checkbox" value="" />我已阅读并同意<span>《众享家用户注册协议》</span>

</ul>

<ul class="line agree">

<input class="registeript" type="button" value="立即注册" />

</ul>

</div>

<div id="foot">

众享家公司copyright@2017-2018

</div>

</body>

</html>

### 4.2.3 CSS代码

body{

*margin*: 0;

*font-size*: 14px;

}

#main{

*width*: 90%;

*margin*: auto;

}

#top{

*height*: 30px;

*border*: 2px solid black;

*line-height*: 30px;

*padding-left*:10px;

}

#nav{

*height*: 60px;

*border*: 2px solid black;

*background*: grey;

}

#center{

*border*:2px solid black;

}

#center ul{

*width*: 80%;

*border*:1px solid black;

*padding*: 0;

*overflow*: hidden;

*height*: 80px;

*margin*: auto;

*list-style-type*: none;

}

#center ul li{

*border*: 1px solid black;

*float*: *left*;

*height*: 80px;

*line-height*: 80px;

}

#center ul li.lab{

*width*: 30%;

*text-align*: *right*;

}

#center ul li.lab b{

*color*: red;

*margin-right*: 3px;

}

#center ul li.ipt{

*width*: 30%;

}

#center ul li.ipt input{

*width*: 260px;

*height*: 36px;

*margin-left*: 10px;

*padding-left*: 10px;

*font-family*: "微软雅黑";

*font-size*: 18px;

}

#center ul li.des{

*width*: 30%;

}

#center ul li.des input{

*width*: 120px;

*height*: 50px;

*border-radius*: 20px;

}

.agree{

*text-align*: center;

*line-height*: 80px;

}

.agree span{

*color*: blue;

}

.agree.registeript{

*width*: 130px;

*height*: 50px;

*border-radius*: 20px;

*background-color*: red;

*font-size*: 20px;

*font-weight*: 800px;

*color*: white;

}

#bottom{

*height*: 50px;

*border*: 2px solid yellow;

}

#foot{

*height*: 50px;

*border*: 2px solid purple;

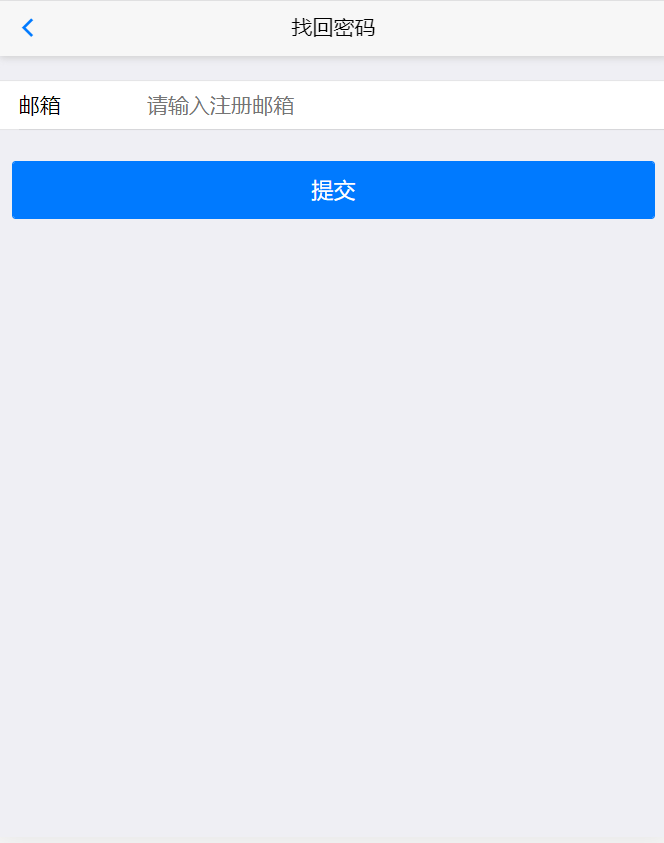
*text-align*: center;

*line-height*: 50px;

}

## 4.3题目3

### 4.3.1 页面运行效果截图



### 4.3.2 HTML代码

<!DOCTYPE html>

<html class="ui-page-login">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width,initial-scale=1,minimum-scale=1,maximum-scale=1,user-scalable=no" />

<title></title>

<link href="css/mui.min.css" rel="stylesheet" />

<link href="css/style.css" rel="stylesheet" />

<style>

.area {

*margin*: 20px auto 0px auto;

}

.mui-input-group:first-child {

*margin-top*: 20px;

}

.mui-input-group label {

*width*: 22%;

}

.mui-input-row label~input,

.mui-input-row label~select,

.mui-input-row label~textarea {

*width*: 78%;

}

.mui-checkbox input[*type*=checkbox],

.mui-radio input[*type*=radio] {

*top*: 6px;

}

.mui-content-padded {

*margin-top*: 25px;

}

.mui-btn {

*padding*: 10px;

}

.mui-input-row label~input, .mui-input-row label~select, .mui-input-row label~textarea{

*margin-top*: 1px;

}

</style>

</head>

<body>

<header class="mui-bar mui-bar-nav">

<a class="mui-action-back mui-icon mui-icon-left-nav mui-pull-left"></a>

<h1 class="mui-title">找回密码</h1>

</header>

<div class="mui-content">

<form class="mui-input-group">

<!--

<div class="mui-input-row">

<label>账号</label>

<input id='account' type="text" class="mui-input-clear mui-input" placeholder="请输入账号">

</div>

-->

<div class="mui-input-row">

<label>邮箱</label>

<input id='email' type="email" class="mui-input-clear mui-input" placeholder="请输入注册邮箱">

</div>

</form>

<div class="mui-content-padded">

<button id='sendMail' class="mui-btn mui-btn-block mui-btn-primary">提交</button>

</div>

</div>

<script src="js/mui.min.js"></script>

<script src="js/app.js"></script>

<script>

(*function*(*$*, *doc*) {

$.init();

$.plusReady(*function*() {

*var* sendButton = doc.getElementById('sendMail');

*var* emailBox = doc.getElementById('email');

sendButton.addEventListener('tap', *function*() {

app.forgetPassword(emailBox.value, *function*(*err*, *info*) {

plus.nativeUI.toast(err || info);

});

}, false);

});

}(mui, *document*));

</script>

</body>

</html>

### 4.3.3 CSS代码

.feedback body {

*background-color*: #EFEFF4;

}

.feedback input,

.feedback textarea {

*border*: none !important;

}

.feedback textarea {

*height*: 100px;

*margin-bottom*: 0 !important;

*padding-bottom*: 0 !important;

}

.feedback .row {

*width*: 100%;

*background-color*: #fff;

}

.feedback p {

*padding*: 10px 15px 0;

}

.feedback button#submit {

*width*: 90%;

*height*: 46px;

*left*: 50%;

*-webkit-transform*: translate(-50%);

}

.feedback .hidden {

*display*: none;

}

.feedback .image-list {

*width*: 100%;

*height*: 85px;

*background-size*: cover;

*padding*: 10px 10px;

*overflow*: hidden;

}

.feedback .image-item {

*width*: 65px;

*height*: 65px;

*background-image*: url(../images/iconfont-tianjia.png);

*background-size*: 100% 100%;

*display*: inline-block;

*position*: relative;

*border-radius*: 5px;

*margin-right*: 10px;

*margin-bottom*: 10px;

*border*: solid 1px #e8e8e8;

}

.feedback .image-item input[*type*="file"] {

*position*: absolute;

*left*: 0px;

*top*: 0px;

*width*: 100%;

*height*: 100%;

*opacity*: 0;

*cursor*: pointer;

*z-index*: 0;

}

.feedback .image-item.space {

*border*: none;

}

.feedback .image-item .image-close {

*position*: absolute;

*display*: inline-block;

*right*: -6px;

*top*: -6px;

*width*: 20px;

*height*: 20px;

*text-align*: center;

*line-height*: 20px;

*border-radius*: 12px;

*background-color*: #FF5053;

*color*: #f3f3f3;

*border*: solid 1px #FF5053;

*font-size*: 9px;

*font-weight*: 200;

*z-index*: 1;

}

.feedback .image-item.space .image-close {

*display*: none;

}

# 五、思考题

1、举例HTML基础标签？举例H5新增标签？

2、web页面设计中，HTML、CSS、JS作用分别是什么？

3、HTML的全称？CSS的全称？（包含英文和中文）。

福建师范大学协和学院《人机交互》实验报告

学号：\_\_\_\_\_\_\_\_ 姓名：\_\_\_\_\_\_\_\_\_\_ 实验日期：\_2019.3.21\_\_ 成绩：\_\_\_\_\_\_\_

实验二 交互系统—数据添加

# 一、实验目的

掌握使用Hbuilder设计并实现前端页面，并将其移植jsp页面；

掌握Eclipse的基本的使用；

掌握tomcat的运行原理；

掌握如何在Eclipse中配置tomcat;

掌握简单的JavaEE项目的环境搭建；

掌握JavaEE web程序的创建方法；

# 二、实验环境（主要仪器设备和软件）

1、操作系统Windows 7

2、HBuilder.

3、Google Chrome

4、tomcat

5、eclipse

6、jdk

# 三、实验内容

1. 创建web工程，命名\*\_net，根据自己的业务命名，同学之间勿重复，例phone\_net；
2. 在webRoot下的lib文件夹下添加mysql驱动；
3. src下面添加package.

com.phone.beans

com.phone.servlets

com.phone.utils

com.phone.daos

1. 在com.phone.utils中添加已经写好的jdbc函数；
2. 创建数据库phone\_db.

创建数据表t\_phone.

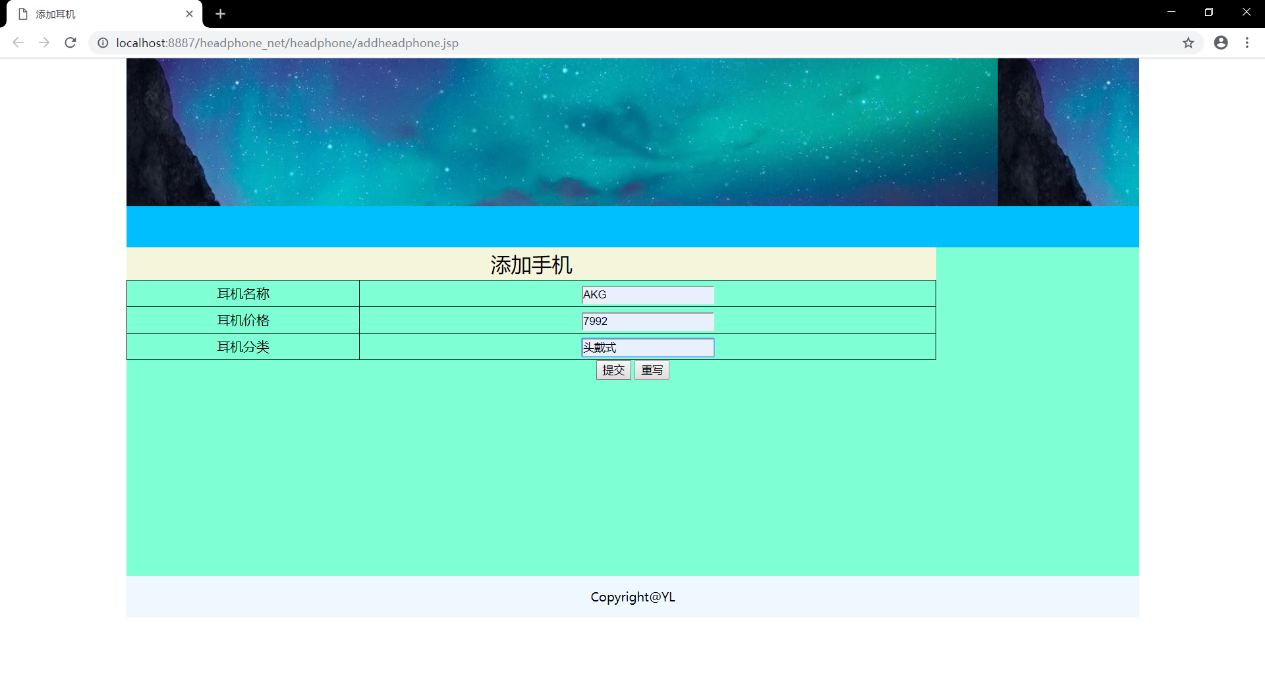
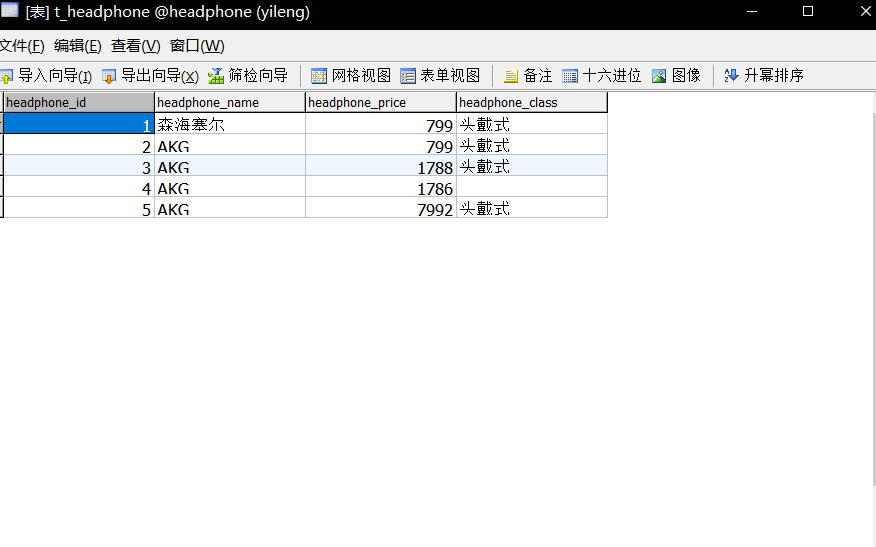
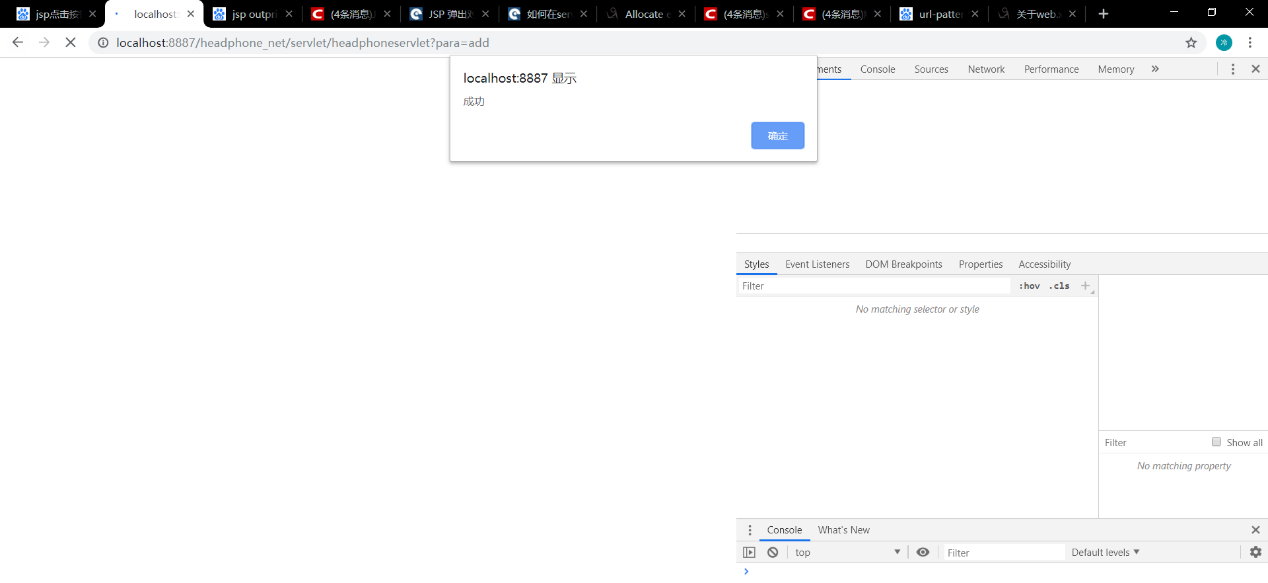
phone\_id,phone\_name,phone\_price ,phone\_class

1. 在com.phone.beans包下创建javabean类：Phone.java
2. com.phone.daos中创建PhoneImpl.java，编写添加函数。
3. 在com.phone.servlets创建PhoneServlet.java文件，编写添加函数。
4. 在web.xml中配置servlet。
5. Hbuilder编写添加页面。
6. 在WebRoot中添加文件夹phone,并在phone中添加addphone.jsp文件。
7. 调试，从页面添加数据到数据库。

# 四、代码及运行结果

## 4.1 页面部分

### 4.1.1 页面运行效果截图



### 4.1.2 HTML代码

<%@ page language="java" contentType="text/html; charset=utf-8"

pageEncoding="utf-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>添加耳机</title>

<link rel="stylesheet" type="text/css" href="../css/headphone/addheadphone.css"/>

</head>

<body>

<div id="all">

<div id="top">

</div>

<div id="nav">

</div>

<div id="main">

<form action="../servlet/headphoneservlet?para=add" method="post">

<table>

<caption>添加手机</caption>

<tr>

<td>耳机名称</td>

<td><input type="text" name="headphone\_name" id="headphone\_name" /></td>

</tr>

<tr>

<td>耳机价格</td>

<td><input type="text" name="headphone\_price" id="headphone\_price" /></td>

</tr>

<tr>

<td>耳机分类</td>

<td><input type="text" name="headphone\_class" id="headphone\_class" /></td>

</tr>

</table>

<div id="button">

<input class="btstyle" type="submit" value="提交" />

<input class="btstyle" type="reset" value="重写" />

</div>

</form>

</div>

<div id="backindex">

<a href="http://localhost:8887/headphone\_net/headphone/index.jsp">

<button>返回首页</button>

</div>

<div id="foot">

Copyright@YL

</div>

</div>

</body>

</html>

## 4.1.3 CSS代码

<%@ page language=*"java"* contentType=*"text/html; charset=utf-8"*

pageEncoding=*"utf-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>添加耳机</title>

<link rel=*"stylesheet"* type=*"text/css"* href=*"../css/headphone/addheadphone.css"*/>

</head>

<body>

<div id=*"all"*>

<div id=*"top"*>

</div>

<div id=*"nav"*>

</div>

<div id=*"main"*>

<form action=*"../servlet/headphoneservlet?para=add"* method=*"post"*>

<table>

<caption>添加手机</caption>

<tr>

<td>耳机名称</td>

<td><input type=*"text"* name=*"headphone\_name"* id=*"headphone\_name"* /></td>

</tr>

<tr>

<td>耳机价格</td>

<td><input type=*"text"* name=*"headphone\_price"* id=*"headphone\_price"* /></td>

</tr>

<tr>

<td>耳机分类</td>

<td><input type=*"text"* name=*"headphone\_class"* id=*"headphone\_class"* /></td>

</tr>

</table>

<div id=*"button"*>

<input class=*"btstyle"* type=*"submit"* value=*"提交"* />

<input class=*"btstyle"* type=*"reset"* value=*"重写"* />

</div>

</form>

</div>

<div id=*"backindex"*>

<a href=*"http://localhost:8887/headphone\_net/headphone/index.jsp"*>

<button>返回首页</button>

</div>

<div id=*"foot"*>

Copyright@YL

</div>

</div>

</body>

</html>

## 4.2 javabean代码

**package** com.head\_phone.beans;

**public** **class** Headphone {

**private** **int** headphone\_id;

**private** String headphone\_name;

**private** **float** headphone\_price;

**private** String headphone\_class;

**public** **int** getHeadphone\_id() {

**return** headphone\_id;

}

**public** **void** setHeadphone\_id(**int** headphone\_id) {

**this**.headphone\_id = headphone\_id;

}

**public** String getHeadphone\_name() {

**return** headphone\_name;

}

**public** **void** setHeadphone\_name(String headphone\_name) {

**this**.headphone\_name = headphone\_name;

}

**public** **float** getHeadphone\_price() {

**return** headphone\_price;

}

**public** **void** setHeadphone\_price(**float** headphone\_price) {

**this**.headphone\_price = headphone\_price;

}

**public** String getHeadphone\_class() {

**return** headphone\_class;

}

**public** **void** setHeadphone\_class(String headphone\_class) {

**this**.headphone\_class = headphone\_class;

}

@Override

**public** String toString() {

**return** "Headphone [headphone\_id=" + headphone\_id + ", headphone\_name=" + headphone\_name + ", headphone\_price="

+ headphone\_price + ", headphone\_class=" + headphone\_class + "]";

}

}

## 4.3 web.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://java.sun.com/xml/ns/javaee"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"* id=*"WebApp\_ID"* version=*"3.0"*>

<display-name>headphone\_net</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>HeadphoneServlet</servlet-name>

<servlet-class>com.headphone\_net.servlet.HeadphoneSerlvet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>HeadphoneServlet</servlet-name>

<url-pattern>/servlet/headphoneservlet</url-pattern>

</servlet-mapping>

## 4.4 dao 代码

package com.headphone\_net.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import javax.xml.transform.Result;

import org.omg.CORBA.PUBLIC\_MEMBER;

import com.head\_phone.beans.Headphone;

import com.headphone\_src.utils.DBUtil;

public class HeadphonedaoImpl {

public boolean modifyHeadphone(Headphone headphone)

{

boolean flag=false;

DBUtil dbUtil=new DBUtil();

Connection conn= (Connection) dbUtil.getConn();

try {

String sql="update t\_headphone set headphone\_name=?,headphone\_price=?,headphone\_class=? where phone\_id=?";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setString(1,headphone.getHeadphone\_name());

pstmt.setFloat(2,headphone.getHeadphone\_price());

pstmt.setString(3,headphone.getHeadphone\_class());

pstmt.setInt(4,headphone.getHeadphone\_id());

int result=pstmt.executeUpdate();

if (1==result) {

flag=true;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

try {

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return flag;

}

public boolean insertheadphone(Headphone headphone)

{

boolean flag=false;

DBUtil dbUtil=new DBUtil();

Connection conn= (Connection) dbUtil.getConn();

try {

String sql="insert into t\_headphone(headphone\_name,headphone\_price,headphone\_class) values(?,?,?)";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setString(1,headphone.getHeadphone\_name());

pstmt.setFloat(2,headphone.getHeadphone\_price());

pstmt.setString(3,headphone.getHeadphone\_class());

int result=pstmt.executeUpdate();

if (1==result) {

flag=true;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

try {

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return flag;

}

//按照id进行查询。

public Headphone searchHeadphoneById(int headphone\_id\_para) {

Headphone headphone=new Headphone();

DBUtil dbUtil=new DBUtil();

Connection conn= (Connection) dbUtil.getConn();

ResultSet rs=null;

try {

String sql="select \* from t\_headphone where headphone\_id=?";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setInt(1, headphone\_id\_para);

rs=pstmt.executeQuery();

while (rs.next()) {

int headphone\_id=rs.getInt("headphone\_id");

String headphone\_name=rs.getString("headphone\_name");

float headphone\_price=rs.getFloat("headphone\_price");

String headphone\_class=rs.getString("headphone\_class");

headphone.setHeadphone\_id(headphone\_id);

headphone.setHeadphone\_name(headphone\_name);

headphone.setHeadphone\_price(headphone\_price);

headphone.setHeadphone\_class(headphone\_class);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

try {

rs.close();

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return headphone;

}

public ArrayList<Headphone> searchHeadPhoneByName(String code\_name\_para){

ArrayList<Headphone> headphonelist=new ArrayList<Headphone>();

DBUtil dbUtil=new DBUtil();

Connection conn= (Connection) dbUtil.getConn();

ResultSet rs=null;

try {

String sql="select \* from t\_headphone where headphone\_name=?";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setString(1, phone\_name\_para);

rs=pstmt.executeQuery();

while (rs.next()) {

int headphone\_id=rs.getInt("headphone\_id");

String headphone\_name=rs.getString("headphone\_name");

float headphone\_price=rs.getFloat("headphone\_price");

String headphone\_class=rs.getString("headphone\_class");

Headphone headphone=new Headphone();

headphone.setHeadphone\_id(headphone\_id);

headphone.setHeadphone\_name(headphone\_name);

headphone.setHeadphone\_price(headphone\_price);

headphone.setHeadphone\_class(headphone\_class);

headphonelist.add(headphone);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

try {

rs.close();

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return headphonelist;

}

public boolean deleteByID(int headphone\_id\_para) {

boolean flag=false;

DBUtil dbUtil=new DBUtil();

Connection conn=dbUtil.getConn();

String sql="delete from t\_headphone where headphone\_id=?";

try {

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setInt(1,headphone\_id\_para);

int result=pstmt.executeUpdate();

if (1==result) {

flag=true;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

try {

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return flag;

}

// public static void main(String[] args) {

// HeadphonedaoImpl headphonedaoImpl=new HeadphonedaoImpl();

// Headphone headphone=new Headphone();

// headphone.setHeadphone\_name("森海塞尔");

// headphone.setHeadphone\_price(799);

// headphone.setHeadphone\_class("头戴式");

// boolean flag=headphonedaoImpl.insertheadphone(headphone);

// System.out.println(flag);

// }

}

## 4.5 servlet代码

package com.headphone\_net.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.ArrayList;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.head\_phone.beans.Headphone;

import com.headphone\_net.dao.HeadphonedaoImpl;

public class HeadphoneSerlvet extends HttpServlet{

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

// TODO Auto-generated method stub

req.setCharacterEncoding("utf-8");

resp.setContentType("text/html;charset=utf-8");

doPost(req, resp);

}

@Override

protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

// TODO Auto-generated method stub

req.setCharacterEncoding("utf-8");

resp.setContentType("text/html;charset=utf-8");

String para=req.getParameter("para");

if (para.equals("add")) {

addheadphone(req,resp);

}else if(para.equals("search")) {

searchHeadphoneByname(req,resp);

}else if (para.equals("delete")) {

deleteHeadphoneByID(req,resp);

}

else if(para.equals("modify1")) {

searchHeadphoneById(req, resp);

}else if(para.equals("modify2")) {

modifyHeadphone(req,resp);

}

// else if (para.equals("searchheadphone\_ajax")) {

// searchheadphone\_ajax(req, resp);

// }

}

private void modifyHeadphone(HttpServletRequest req, HttpServletResponse resp) {

// TODO Auto-generated method stub

String headphone\_id\_str=req.getParameter("headphone\_id");

int headphone\_id=Integer.parseInt(headphone\_id\_str);

String headphone\_name=req.getParameter("headphone\_name");

String headphone\_price\_str=req.getParameter("headphone\_price");

float headphone\_price=Float.parseFloat(headphone\_price\_str);

String headphone\_class=req.getParameter("headphone\_class");

Headphone headphone=new Headphone();

headphone.setHeadphone\_id(headphone\_id);

headphone.setHeadphone\_name(headphone\_name);

headphone.setHeadphone\_price(headphone\_price);

headphone.setHeadphone\_class(headphone\_class);

HeadphonedaoImpl headphonedaoImpl=new HeadphonedaoImpl();

boolean flag=headphonedaoImpl.modifyHeadphone(headphone);

try {

PrintWriter out = resp.getWriter();

// if (flag) {

// out.println("<SCRIPT LANGUAGE='JavaScript'>alert('添加成功！')JavaScript:history.back(-1);</SCRIPT>");

// } else {

// out.println("<SCRIPT LANGUAGE='JavaScript'>alert('添加失败！')JavaScript:history.back(-1);</SCRIPT>");

// }

if (flag) {

out.print("<script>alert('修改成功'); </script>");

} else {

out.print("<script>alert('修改失败'); </script>");

}

} catch (Exception e) {

// TODO: handle exception

}

}

private void searchHeadphoneById(HttpServletRequest req, HttpServletResponse resp) {

// TODO Auto-generated method stub

String headphone\_id\_str=req.getParameter("headphone\_id");

int headphone\_id=Integer.parseInt(headphone\_id\_str);

HeadphonedaoImpl headphonedaoImpl=new HeadphonedaoImpl();

Headphone headphone=headphonedaoImpl.searchHeadphoneById(headphone\_id);

try {

req.setAttribute("headphone", headphone);

req.getRequestDispatcher("/headphone/modifyheadphone.jsp").forward(req, resp);

} catch (ServletException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

// private void searchheadphone\_ajax(HttpServletRequest req, HttpServletResponse resp) {

// // TODO Auto-generated method stub

// System.out.println("6666666");

// try {

// String headphone\_name=req.getParameter("headphone\_name");

// HeadphonedaoImpl headphonedaoImpl=new HeadphonedaoImpl();

// ArrayList<Headphone>headphonelist=headphonedaoImpl.searchHeadPhoneByName(headphone\_name);

//

// PrintWriter out=resp.getWriter();

// out.println("ajax-success");

// out.flush();

// out.close();

// } catch (IOException e) {

// // TODO Auto-generated catch block

// e.printStackTrace();

// }

//

// }

private void deleteHeadphoneByID(HttpServletRequest req, HttpServletResponse resp) {

// TODO Auto-generated method stub

String headphone\_id\_str=req.getParameter("headphone\_id");

int headphone\_id=Integer.parseInt(headphone\_id\_str);

HeadphonedaoImpl headphonedaoImpl=new HeadphonedaoImpl();

boolean flag=headphonedaoImpl.deleteByID(headphone\_id);

try {

PrintWriter out = resp.getWriter();

if (flag) {

out.print("<script>alert('删除成功'); </script>");

} else {

out.print("<script>alert('删除失败'); </script>");

}

} catch (Exception e) {

// TODO: handle exception

}

}

public void searchHeadphoneByname(HttpServletRequest req, HttpServletResponse resp) {

// TODO Auto-generated method stub

String headphone\_name=req.getParameter("headphone\_name");

HeadphonedaoImpl headphonedaoImpl=new HeadphonedaoImpl();

ArrayList<Headphone>headphonelist=headphonedaoImpl.searchHeadPhoneByName(headphone\_name);

try {

req.setAttribute("headphonelist", headphonelist);

req.getRequestDispatcher("/headphone/searchheadphone.jsp").forward(req, resp);

} catch (ServletException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public void addheadphone(HttpServletRequest req, HttpServletResponse resp) {

String headphone\_name=req.getParameter("headphone\_name");

String headphone\_price\_str=req.getParameter("headphone\_price");

float headphone\_price=Float.parseFloat(headphone\_price\_str);

String headphone\_class=req.getParameter("headphone\_class");

Headphone headphone=new Headphone();

headphone.setHeadphone\_name(headphone\_name);

headphone.setHeadphone\_price(headphone\_price);

headphone.setHeadphone\_class(headphone\_class);

HeadphonedaoImpl headphonedaoImpl=new HeadphonedaoImpl();

boolean flag=headphonedaoImpl.insertheadphone(headphone);

try {

PrintWriter out = resp.getWriter();

// if (flag) {

// out.println("<SCRIPT LANGUAGE='JavaScript'>alert('添加成功！')JavaScript:history.back(-1);</SCRIPT>");

// } else {

// out.println("<SCRIPT LANGUAGE='JavaScript'>alert('添加失败！')JavaScript:history.back(-1);</SCRIPT>");

// }

if (flag) {

out.print("<script>alert('添加成功'); </script>");

} else {

out.print("<script>alert('添加失败'); </script>");

}

} catch (Exception e) {

// TODO: handle exception

}

}

}

## 4.6 数据库与数据表截图

# 五、思考题

1、JavaEE的主要研究内容？

2、Tomcat在项目中的主要作用？

3、什么是JSP?JSP在项目中的作用？

福建师范大学协和学院《人机交互》实验报告

学号：\_\_\_\_\_\_\_\_ 姓名：\_\_\_\_\_\_\_\_\_\_ 实验日期：\_2019.4.4\_\_ 成绩：\_\_\_\_\_\_\_

实验三 交互系统—数据查询删除

# 一、实验目的

掌握编写Servlet的步骤；

（1）、继承HttpServlet类；

（2）、重写HttpServlet的doGet()方法；

（3）、配置Servlet方法；

（4）、部署Servlet；

（5）、测试Servlet。

掌握Servlet生命周期：

（1）、该容器调用 init() 方法；

（2）、请求 Servlet，则容器调用 service() 方法，然后调用doGet()方法，调用doPost()方法；

（3）、销毁实例之前调用 destroy() 方法；

掌握MVC框架的执行流程；

掌握JavaEE中jdbc的API的用法。

# 二、实验环境（主要仪器设备和软件）

1、操作系统Windows 7

2、HBuilder.

3、Google Chrome

4、tomcat

5、eclipse

6、jdk

# 三、实验内容

（1）、HBuilder上设计searchphone.html页面；

（2）、将searchphone.html移植到eclipse的webRoot下面的searchphone.jsp页面；

（3）、编写Dao的查询代码，编写searchDao(String phone\_name)函数；

（4）、编写Servlet的代码，编写servlet的searchDao(request,response)代码；

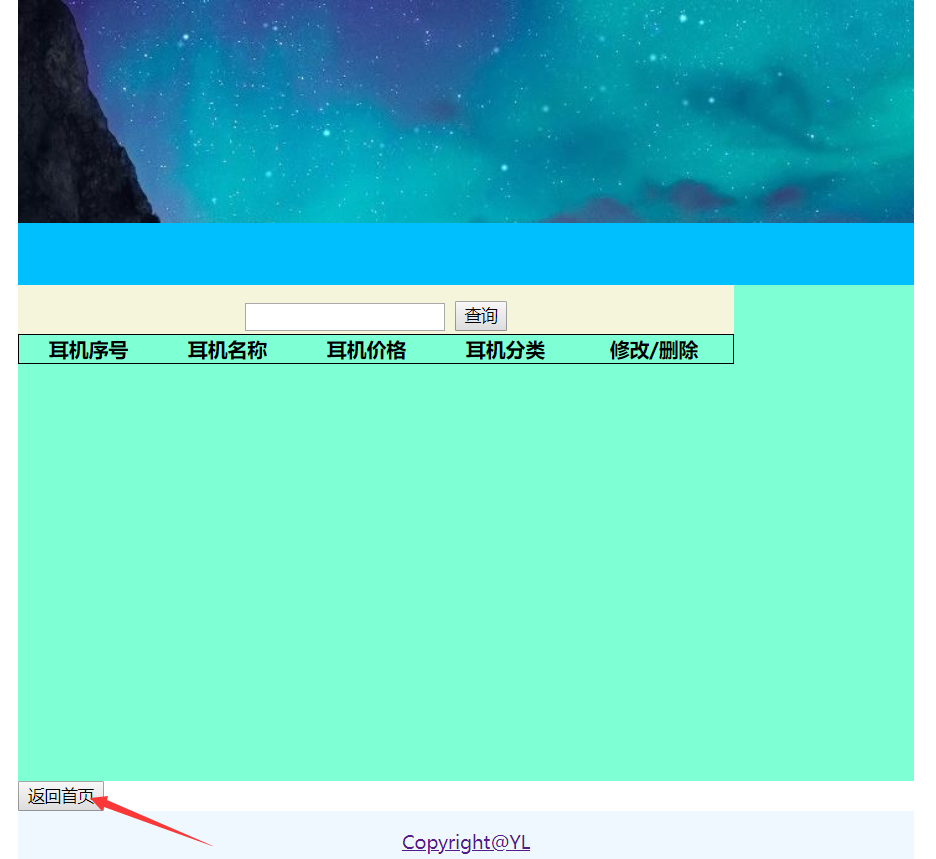
（5）、联合调试；

（6）、编写删除代码。

# 四、代码及运行结果

## 4.1 查询页面部分

### 4.1.1 页面运行效果截图



### 4.1.2 HTML代码

<%@page import="org.apache.jasper.tagplugins.jstl.core.If"%>

<%@page import="com.head\_phone.beans.\*"%>

<%@page import="java.util.\*"%>

<%@ page language="java" contentType="text/html; charset=utf-8"

pageEncoding="utf-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>查找耳机</title>

<link rel="stylesheet" type="text/css" href="../css/headphone/addheadphone.css"/>

</head>

<body>

<div id="all">

<div id="top">

</div>

<div id="nav">

</div>

<div id="main">

<form action="../servlet/headphoneservlet?para=search" method="post">

<table>

<caption>

<input type="text" name="headphone\_name" id="headphone\_name" />

<input type="submit" value="查询" />

</caption>

<tr>

<th>耳机序号</th>

<th>耳机名称</th>

<th>耳机价格</th>

<th>耳机分类</th>

<th>修改/删除</th>

</tr>

<%

ArrayList<Headphone>headphonelist=(ArrayList<Headphone>)request.getAttribute("headphonelist");

**if**(headphonelist!=**null**){

**for**(**int** i=0;i<headphonelist.size();i++)

{

Headphone headphone=headphonelist.get(i);

**int** headphone\_id=headphone.getHeadphone\_id();

String headphone\_name=headphone.getHeadphone\_name();

**float** headphone\_price=headphone.getHeadphone\_price();

String headphone\_class=headphone.getHeadphone\_class();

%>

<tr>

<td><%=headphone\_id %></td>

<td><%=headphone\_name %></td>

<td><%=headphone\_price %></td>

<td><%=headphone\_class %></td>

<td>

<a href="../servlet/headphoneservlet?para=modify1&headphone\_id=<%=headphone\_id %>">修改</a>

<a href="../servlet/headphoneservlet?para=delete&headphone\_id=<%=headphone\_id %>">删除</a>

</td>

</tr>

<%

}

}

%>

</table>

</form>

</div>

<a href="http://localhost:8887/headphone\_net/headphone/index.jsp">

<button>返回首页</button>

<div id="foot">

Copyright@YL

</div>

</div>

</body>

</html>

### 4.1.3 CSS代码

body{

margin: *0*;

}

*#all*{

width: *80%*;

height: *600px*;

/\*border: 1px solid black ;\*/

margin: *auto*;

}

*#all* *#top*{

height: *180px*;

background-image: *url(../../img/headphone/topbg.JPG)*;

}

*#nav*{

height: *50px*;

background: *deepskyblue*;

}

*#main*{

height: *400px*;

background-color: *aquamarine*

}

*#main* table{

border: *1px* *solid* *black*;

width: *80%*;

border-collapse: *collapse*;

}

*#main* table caption{

background: *beige*;

line-height: *40px*;

font-size: *25px*;

font-family: *"寰蒋闆呴粦"*;

}

*#main* table td{

border:*1px* *solid* *black*;

line-height: *30px*;

text-align: *center*;

}

*#button*{

text-align:*center*;

}

*#foot*{

height: *50px*;

background: *aliceblue*;

font-size: *15px*;

font-family: *"微软雅黑"*;

line-height: *50px*;

text-align:*center*;

}

## 4.2 查询函数dao代码

**public** Headphone searchHeadphoneById(**int** headphone\_id\_para) {

Headphone headphone=**new** Headphone();

DBUtil dbUtil=**new** DBUtil();

Connection conn= (Connection) dbUtil.getConn();

ResultSet rs=**null**;

**try** {

String sql="select \* from t\_headphone where headphone\_id=?";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setInt(1, headphone\_id\_para);

rs=pstmt.executeQuery();

**while** (rs.next()) {

**int** headphone\_id=rs.getInt("headphone\_id");

String headphone\_name=rs.getString("headphone\_name");

**float** headphone\_price=rs.getFloat("headphone\_price");

String headphone\_class=rs.getString("headphone\_class");

headphone.setHeadphone\_id(headphone\_id);

headphone.setHeadphone\_name(headphone\_name);

headphone.setHeadphone\_price(headphone\_price);

headphone.setHeadphone\_class(headphone\_class);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}**finally** {

**try** {

rs.close();

conn.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**return** headphone;

}

**public** ArrayList<Headphone> searchHeadPhoneByName(String code\_name\_para){

ArrayList<Headphone> headphonelist=**new** ArrayList<Headphone>();

DBUtil dbUtil=**new** DBUtil();

Connection conn= (Connection) dbUtil.getConn();

ResultSet rs=**null**;

**try** {

String sql="select \* from t\_headphone where headphone\_name=?";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setString(1, phone\_name\_para);

rs=pstmt.executeQuery();

**while** (rs.next()) {

**int** headphone\_id=rs.getInt("headphone\_id");

String headphone\_name=rs.getString("headphone\_name");

**float** headphone\_price=rs.getFloat("headphone\_price");

String headphone\_class=rs.getString("headphone\_class");

Headphone headphone=**new** Headphone();

headphone.setHeadphone\_id(headphone\_id);

headphone.setHeadphone\_name(headphone\_name);

headphone.setHeadphone\_price(headphone\_price);

headphone.setHeadphone\_class(headphone\_class);

headphonelist.add(headphone);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}**finally** {

**try** {

rs.close();

conn.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**return** headphonelist;

}

## 4.4查询函数servlet代码

package com.headphone\_net.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.ArrayList;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.head\_phone.beans.Headphone;

import com.headphone\_net.dao.HeadphonedaoImpl;

public class HeadphoneSerlvet extends HttpServlet{

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

// TODO Auto-generated method stub

req.setCharacterEncoding("utf-8");

resp.setContentType("text/html;charset=utf-8");

doPost(req, resp);

}

@Override

protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

// TODO Auto-generated method stub

req.setCharacterEncoding("utf-8");

resp.setContentType("text/html;charset=utf-8");

String para=req.getParameter("para");

if (para.equals("add")) {

addheadphone(req,resp);

}else if(para.equals("search")) {

searchHeadphoneByname(req,resp);

}else if (para.equals("delete")) {

deleteHeadphoneByID(req,resp);

}

else if(para.equals("modify1")) {

searchHeadphoneById(req, resp);

}else if(para.equals("modify2")) {

modifyHeadphone(req,resp);

}

// else if (para.equals("searchheadphone\_ajax")) {

// searchheadphone\_ajax(req, resp);

// }

}

private void modifyHeadphone(HttpServletRequest req, HttpServletResponse resp) {

// TODO Auto-generated method stub

String headphone\_id\_str=req.getParameter("headphone\_id");

int headphone\_id=Integer.parseInt(headphone\_id\_str);

String headphone\_name=req.getParameter("headphone\_name");

String headphone\_price\_str=req.getParameter("headphone\_price");

float headphone\_price=Float.parseFloat(headphone\_price\_str);

String headphone\_class=req.getParameter("headphone\_class");

Headphone headphone=new Headphone();

headphone.setHeadphone\_id(headphone\_id);

headphone.setHeadphone\_name(headphone\_name);

headphone.setHeadphone\_price(headphone\_price);

headphone.setHeadphone\_class(headphone\_class);

HeadphonedaoImpl headphonedaoImpl=new HeadphonedaoImpl();

boolean flag=headphonedaoImpl.modifyHeadphone(headphone);

try {

PrintWriter out = resp.getWriter();

// if (flag) {

// out.println("<SCRIPT LANGUAGE='JavaScript'>alert('添加成功！')JavaScript:history.back(-1);</SCRIPT>");

// } else {

// out.println("<SCRIPT LANGUAGE='JavaScript'>alert('添加失败！')JavaScript:history.back(-1);</SCRIPT>");

// }

if (flag) {

out.print("<script>alert('修改成功'); </script>");

} else {

out.print("<script>alert('修改失败'); </script>");

}

} catch (Exception e) {

// TODO: handle exception

}

}

private void searchHeadphoneById(HttpServletRequest req, HttpServletResponse resp) {

// TODO Auto-generated method stub

String headphone\_id\_str=req.getParameter("headphone\_id");

int headphone\_id=Integer.parseInt(headphone\_id\_str);

HeadphonedaoImpl headphonedaoImpl=new HeadphonedaoImpl();

Headphone headphone=headphonedaoImpl.searchHeadphoneById(headphone\_id);

try {

req.setAttribute("headphone", headphone);

req.getRequestDispatcher("/headphone/modifyheadphone.jsp").forward(req, resp);

} catch (ServletException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

// private void searchheadphone\_ajax(HttpServletRequest req, HttpServletResponse resp) {

// // TODO Auto-generated method stub

// System.out.println("6666666");

// try {

// String headphone\_name=req.getParameter("headphone\_name");

// HeadphonedaoImpl headphonedaoImpl=new HeadphonedaoImpl();

// ArrayList<Headphone>headphonelist=headphonedaoImpl.searchHeadPhoneByName(headphone\_name);

//

// PrintWriter out=resp.getWriter();

// out.println("ajax-success");

// out.flush();

// out.close();

// } catch (IOException e) {

// // TODO Auto-generated catch block

// e.printStackTrace();

// }

//

// }

private void deleteHeadphoneByID(HttpServletRequest req, HttpServletResponse resp) {

// TODO Auto-generated method stub

String headphone\_id\_str=req.getParameter("headphone\_id");

int headphone\_id=Integer.parseInt(headphone\_id\_str);

HeadphonedaoImpl headphonedaoImpl=new HeadphonedaoImpl();

boolean flag=headphonedaoImpl.deleteByID(headphone\_id);

try {

PrintWriter out = resp.getWriter();

if (flag) {

out.print("<script>alert('删除成功'); </script>");

} else {

out.print("<script>alert('删除失败'); </script>");

}

} catch (Exception e) {

// TODO: handle exception

}

}

public void searchHeadphoneByname(HttpServletRequest req, HttpServletResponse resp) {

// TODO Auto-generated method stub

String headphone\_name=req.getParameter("headphone\_name");

HeadphonedaoImpl headphonedaoImpl=new HeadphonedaoImpl();

ArrayList<Headphone>headphonelist=headphonedaoImpl.searchHeadPhoneByName(headphone\_name);

try {

req.setAttribute("headphonelist", headphonelist);

req.getRequestDispatcher("/headphone/searchheadphone.jsp").forward(req, resp);

} catch (ServletException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public void addheadphone(HttpServletRequest req, HttpServletResponse resp) {

String headphone\_name=req.getParameter("headphone\_name");

String headphone\_price\_str=req.getParameter("headphone\_price");

float headphone\_price=Float.parseFloat(headphone\_price\_str);

String headphone\_class=req.getParameter("headphone\_class");

Headphone headphone=new Headphone();

headphone.setHeadphone\_name(headphone\_name);

headphone.setHeadphone\_price(headphone\_price);

headphone.setHeadphone\_class(headphone\_class);

HeadphonedaoImpl headphonedaoImpl=new HeadphonedaoImpl();

boolean flag=headphonedaoImpl.insertheadphone(headphone);

try {

PrintWriter out = resp.getWriter();

// if (flag) {

// out.println("<SCRIPT LANGUAGE='JavaScript'>alert('添加成功！')JavaScript:history.back(-1);</SCRIPT>");

// } else {

// out.println("<SCRIPT LANGUAGE='JavaScript'>alert('添加失败！')JavaScript:history.back(-1);</SCRIPT>");

// }

if (flag) {

out.print("<script>alert('添加成功'); </script>");

} else {

out.print("<script>alert('添加失败'); </script>");

}

} catch (Exception e) {

// TODO: handle exception

}

}

}

## 4.5删除函数dao代码

package com.headphone\_net.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import javax.xml.transform.Result;

import org.omg.CORBA.PUBLIC\_MEMBER;

import com.head\_phone.beans.Headphone;

import com.headphone\_src.utils.DBUtil;

public class HeadphonedaoImpl {

public boolean modifyHeadphone(Headphone headphone)

{

boolean flag=false;

DBUtil dbUtil=new DBUtil();

Connection conn= (Connection) dbUtil.getConn();

try {

String sql="update t\_headphone set headphone\_name=?,headphone\_price=?,headphone\_class=? where phone\_id=?";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setString(1,headphone.getHeadphone\_name());

pstmt.setFloat(2,headphone.getHeadphone\_price());

pstmt.setString(3,headphone.getHeadphone\_class());

pstmt.setInt(4,headphone.getHeadphone\_id());

int result=pstmt.executeUpdate();

if (1==result) {

flag=true;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

try {

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return flag;

}

public boolean insertheadphone(Headphone headphone)

{

boolean flag=false;

DBUtil dbUtil=new DBUtil();

Connection conn= (Connection) dbUtil.getConn();

try {

String sql="insert into t\_headphone(headphone\_name,headphone\_price,headphone\_class) values(?,?,?)";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setString(1,headphone.getHeadphone\_name());

pstmt.setFloat(2,headphone.getHeadphone\_price());

pstmt.setString(3,headphone.getHeadphone\_class());

int result=pstmt.executeUpdate();

if (1==result) {

flag=true;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

try {

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return flag;

}

//按照id进行查询。

public Headphone searchHeadphoneById(int headphone\_id\_para) {

Headphone headphone=new Headphone();

DBUtil dbUtil=new DBUtil();

Connection conn= (Connection) dbUtil.getConn();

ResultSet rs=null;

try {

String sql="select \* from t\_headphone where headphone\_id=?";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setInt(1, headphone\_id\_para);

rs=pstmt.executeQuery();

while (rs.next()) {

int headphone\_id=rs.getInt("headphone\_id");

String headphone\_name=rs.getString("headphone\_name");

float headphone\_price=rs.getFloat("headphone\_price");

String headphone\_class=rs.getString("headphone\_class");

headphone.setHeadphone\_id(headphone\_id);

headphone.setHeadphone\_name(headphone\_name);

headphone.setHeadphone\_price(headphone\_price);

headphone.setHeadphone\_class(headphone\_class);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

try {

rs.close();

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return headphone;

}

public ArrayList<Headphone> searchHeadPhoneByName(String code\_name\_para){

ArrayList<Headphone> headphonelist=new ArrayList<Headphone>();

DBUtil dbUtil=new DBUtil();

Connection conn= (Connection) dbUtil.getConn();

ResultSet rs=null;

try {

String sql="select \* from t\_headphone where headphone\_name=?";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setString(1, phone\_name\_para);

rs=pstmt.executeQuery();

while (rs.next()) {

int headphone\_id=rs.getInt("headphone\_id");

String headphone\_name=rs.getString("headphone\_name");

float headphone\_price=rs.getFloat("headphone\_price");

String headphone\_class=rs.getString("headphone\_class");

Headphone headphone=new Headphone();

headphone.setHeadphone\_id(headphone\_id);

headphone.setHeadphone\_name(headphone\_name);

headphone.setHeadphone\_price(headphone\_price);

headphone.setHeadphone\_class(headphone\_class);

headphonelist.add(headphone);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

try {

rs.close();

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return headphonelist;

}

public boolean deleteByID(int headphone\_id\_para) {

boolean flag=false;

DBUtil dbUtil=new DBUtil();

Connection conn=dbUtil.getConn();

String sql="delete from t\_headphone where headphone\_id=?";

try {

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setInt(1,headphone\_id\_para);

int result=pstmt.executeUpdate();

if (1==result) {

flag=true;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

try {

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return flag;

}

// public static void main(String[] args) {

// HeadphonedaoImpl headphonedaoImpl=new HeadphonedaoImpl();

// Headphone headphone=new Headphone();

// headphone.setHeadphone\_name("森海塞尔");

// headphone.setHeadphone\_price(799);

// headphone.setHeadphone\_class("头戴式");

// boolean flag=headphonedaoImpl.insertheadphone(headphone);

// System.out.println(flag);

// }

}

## 4.6删除函数servlet代码

**private** **void** deleteHeadphoneByID(HttpServletRequest req, HttpServletResponse resp) {

// **TODO** Auto-generated method stub

String headphone\_id\_str=req.getParameter("headphone\_id");

**int** headphone\_id=Integer.*parseInt*(headphone\_id\_str);

HeadphonedaoImpl headphonedaoImpl=**new** HeadphonedaoImpl();

**boolean** flag=headphonedaoImpl.deleteByID(headphone\_id);

**try** {

PrintWriter out = resp.getWriter();

**if** (flag) {

out.print("<script>alert('删除成功'); </script>");

} **else** {

out.print("<script>alert('删除失败'); </script>");

}

} **catch** (Exception e) {

// **TODO**: handle exception

}

}

# 五、思考题

1、在JavaEE项目中，MVC架构中的M、V、C分别指的什么？

2、JavaEE中JDBC中的Statement类和PreparedStatement类的区别？

3、JavaEE中，servlet的编写步骤是什么？

福建师范大学协和学院《人机交互》实验报告

学号：\_\_\_\_\_\_\_\_ 姓名：\_\_\_\_\_\_\_\_\_\_ 实验日期：\_2019.4.18\_\_ 成绩：\_\_\_\_\_\_\_

实验四 交互系统—数据修改

# 一、实验目的

掌握form表单提交的post方法和get方法的区别；

掌握转发和重定向的区别；

掌握jdbc代码中异常的使用；

掌握jsp表达式的使用；

掌握javabean的编写。

# 二、实验环境（主要仪器设备和软件）

1、操作系统Windows 7

2、HBuilder.

3、Google Chrome

4、tomcat

5、eclipse

6、jdk

# 三、实验内容

修改功能的主要思路：查询页面点击修改，根据id从数据库中查询出一条数据，并填充到修改页面，再重新提交修改页面的url。

（1）、使用Hbuilder设计修改页面；

（2）、创建修改页面的jsp页面；

（3）、在查询页面，点击修改，查询一条记录，填充到修改页面，

编写相应的dao和servlet代码；

（4）、编写修改dao类中的代码；

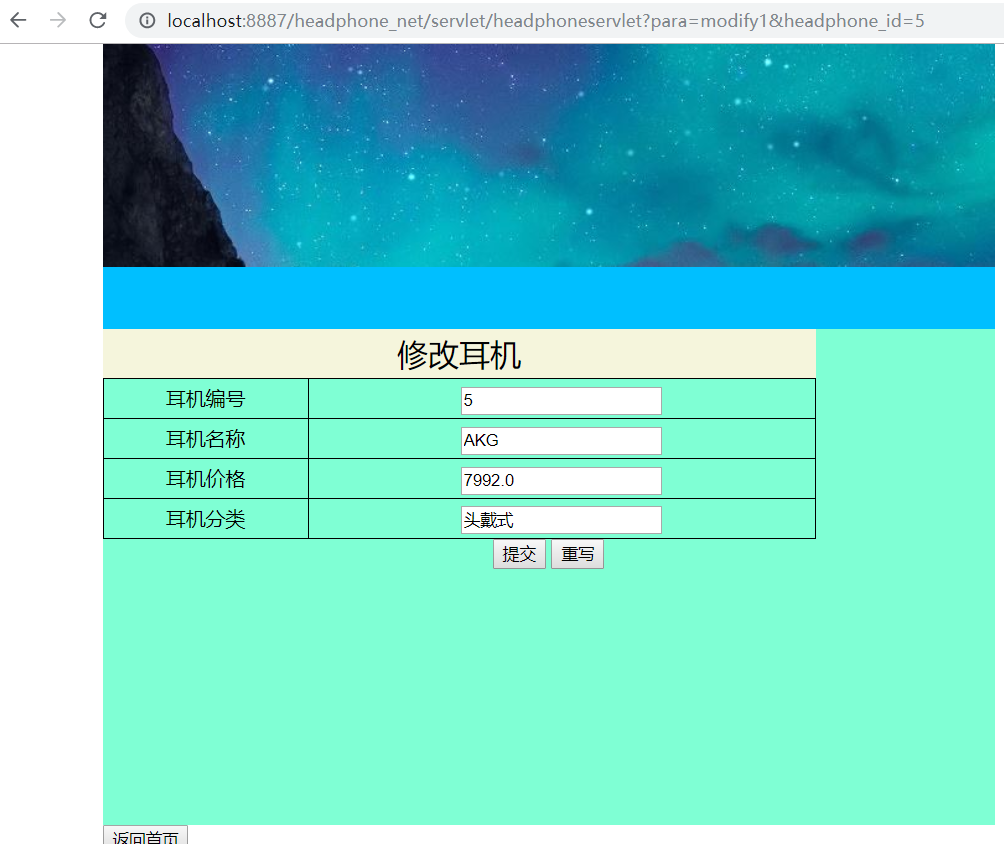
（5）、编写修改servlet类中的代码；

（6）、联合调试。

# 四、代码及运行结果

## 4.1 查询页面部分

### 4.1.1 页面运行效果截图



### 4.1.2 修改的HTML代码

<%@page import="com.head\_phone.beans.Headphone"%>

<%@ page language="java" contentType="text/html; charset=utf-8"

pageEncoding="utf-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>添加耳机</title>

<link rel="stylesheet" type="text/css" href="../css/headphone/addheadphone.css"/>

</head>

<body>

<div id="all">

<div id="top">

</div>

<div id="nav">

</div>

<div id="main">

<form action="../servlet/headphoneservlet?para=add" method="post">

<table>

<%

Headphone headphone=(Headphone)request.getAttribute("headphone");

**int** headphone\_id=headphone.getHeadphone\_id();

String headphone\_name=headphone.getHeadphone\_name();

**float** headphone\_price=headphone.getHeadphone\_price();

String headphone\_class=headphone.getHeadphone\_class();

%>

<caption>修改耳机</caption>

<tr>

<td>耳机编号</td>

<td><input type="text" name="headphone\_id" id="headphone\_id" value="<%=headphone\_id%>" readonly="readonly" /></td>

</tr>

<tr>

<td>耳机名称</td>

<td><input type="text" name="headphone\_name" id="headphone\_name" value="<%=headphone\_name%>" /></td>

</tr>

<tr>

<td>耳机价格</td>

<td><input type="text" name="headphone\_price" id="headphone\_price" value="<%=headphone\_price%>" /></td>

</tr>

<tr>

<td>耳机分类</td>

<td><input type="text" name="headphone\_class" id="headphone\_class" value="<%=headphone\_class%>" /></td>

</tr>

</table>

<div id="button">

<input class="btstyle" type="submit" value="提交" />

<input class="btstyle" type="reset" value="重写" />

</div>

</form>

</div>

<div id="backindex">

<a href="http://localhost:8887/headphone\_net/headphone/index.jsp">

<button>返回首页</button>

</div>

<div id="foot">

Copyright@YL

</div>

</div>

</body>

</html>

## 4.2 按id查询的dao代码

**public** ArrayList<Headphone> searchHeadPhoneByName(String code\_name\_para){

ArrayList<Headphone> headphonelist=**new** ArrayList<Headphone>();

DBUtil dbUtil=**new** DBUtil();

Connection conn= (Connection) dbUtil.getConn();

ResultSet rs=**null**;

**try** {

String sql="select \* from t\_headphone where headphone\_name=?";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setString(1, phone\_name\_para);

rs=pstmt.executeQuery();

**while** (rs.next()) {

**int** headphone\_id=rs.getInt("headphone\_id");

String headphone\_name=rs.getString("headphone\_name");

**float** headphone\_price=rs.getFloat("headphone\_price");

String headphone\_class=rs.getString("headphone\_class");

Headphone headphone=**new** Headphone();

headphone.setHeadphone\_id(headphone\_id);

headphone.setHeadphone\_name(headphone\_name);

headphone.setHeadphone\_price(headphone\_price);

headphone.setHeadphone\_class(headphone\_class);

headphonelist.add(headphone);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}**finally** {

**try** {

rs.close();

conn.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**return** headphonelist;

}

## 4.3按id查询servlet代码

**public** Headphone searchHeadphoneById(**int** headphone\_id\_para) {

Headphone headphone=**new** Headphone();

DBUtil dbUtil=**new** DBUtil();

Connection conn= (Connection) dbUtil.getConn();

ResultSet rs=**null**;

**try** {

String sql="select \* from t\_headphone where headphone\_id=?";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setInt(1, headphone\_id\_para);

rs=pstmt.executeQuery();

**while** (rs.next()) {

**int** headphone\_id=rs.getInt("headphone\_id");

String headphone\_name=rs.getString("headphone\_name");

**float** headphone\_price=rs.getFloat("headphone\_price");

String headphone\_class=rs.getString("headphone\_class");

headphone.setHeadphone\_id(headphone\_id);

headphone.setHeadphone\_name(headphone\_name);

headphone.setHeadphone\_price(headphone\_price);

headphone.setHeadphone\_class(headphone\_class);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}**finally** {

**try** {

rs.close();

conn.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**return** headphone;

}

## 4.4 修改函数dao代码

**public** **boolean** modifyHeadphone(Headphone headphone)

{

**boolean** flag=**false**;

DBUtil dbUtil=**new** DBUtil();

Connection conn= (Connection) dbUtil.getConn();

**try** {

String sql="update t\_headphone set headphone\_name=?,headphone\_price=?,headphone\_class=? where phone\_id=?";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setString(1,headphone.getHeadphone\_name());

pstmt.setFloat(2,headphone.getHeadphone\_price());

pstmt.setString(3,headphone.getHeadphone\_class());

pstmt.setInt(4,headphone.getHeadphone\_id());

**int** result=pstmt.executeUpdate();

**if** (1==result) {

flag=**true**;

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}**finally** {

**try** {

conn.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**return** flag;

}

## 4.5 修改函数servlet代码

**private** **void** modifyHeadphone(HttpServletRequest req, HttpServletResponse resp) {

// **TODO** Auto-generated method stub

String headphone\_id\_str=req.getParameter("headphone\_id");

**int** headphone\_id=Integer.*parseInt*(headphone\_id\_str);

String headphone\_name=req.getParameter("headphone\_name");

String headphone\_price\_str=req.getParameter("headphone\_price");

**float** headphone\_price=Float.*parseFloat*(headphone\_price\_str);

String headphone\_class=req.getParameter("headphone\_class");

Headphone headphone=**new** Headphone();

headphone.setHeadphone\_id(headphone\_id);

headphone.setHeadphone\_name(headphone\_name);

headphone.setHeadphone\_price(headphone\_price);

headphone.setHeadphone\_class(headphone\_class);

HeadphonedaoImpl headphonedaoImpl=**new** HeadphonedaoImpl();

**boolean** flag=headphonedaoImpl.modifyHeadphone(headphone);

**try** {

PrintWriter out = resp.getWriter();

// if (flag) {

// out.println("<SCRIPT LANGUAGE='JavaScript'>alert('添加成功！')JavaScript:history.back(-1);</SCRIPT>");

// } else {

// out.println("<SCRIPT LANGUAGE='JavaScript'>alert('添加失败！')JavaScript:history.back(-1);</SCRIPT>");

// }

**if** (flag) {

out.print("<script>alert('修改成功'); </script>");

} **else** {

out.print("<script>alert('修改失败'); </script>");

}

} **catch** (Exception e) {

// **TODO**: handle exception

}

}

# 五、思考题

1、java web程序设计中，request对象的主要作用是什么？

2、java web程序设计中，response对象的主要作用是什么？

3、form表单提交的post方法和get方法的区别是什么？

福建师范大学协和学院《人机交互》实验报告

学号：\_\_\_\_\_\_\_\_ 姓名：\_\_\_\_\_\_\_\_\_\_ 实验日期：\_2019.5.02 成绩：\_\_\_\_\_\_\_

实验五 交互系统—ajax交互

# 一、实验目的

掌握ajax的定义：Asynchronous JavaScript and XML（异步的 JavaScript 和 XML）。

掌握ajax对象创建方法

var xmlhttp;

if (window.XMLHttpRequest)

{// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp=new XMLHttpRequest();

else

{// code for IE6, IE5

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

掌握ajax的特点：

AJAX 不是新的编程语言，而是一种使用现有标准的新方法；

AJAX 是与服务器交换数据并更新部分网页的艺术，在不重新加载整个页面的情况下。

# 二、实验环境（主要仪器设备和软件）

1、操作系统Windows 7

2、HBuilder.

3、Google Chrome

4、tomcat

5、eclipse

6、jdk

# 三、实验内容

（1）设计并实现一个图片轮播。

a)编写函数image\_go\_1():

循环每秒钟在控制台打印一个数字，1,2,3,4,5,6.

b)编写函数image\_go\_2():

循环每秒钟使每个label的背景色改颜色。

c) 编写函数image\_go\_3():

循环每2秒钟同时使图片得到改变。

（2）设计一个form表单，添加按钮，submit提交。

（3）设计一个form表单，添加按钮，ajax提交。

（4）添加ajax的servlet类；

（5）在web.xml中配置ajax模块的servlet。

（6）完成普通提交的服务器端代码编写（服务器返回一句话）；

（7）完成ajax提交的服务器端的代码编写（服务器返回一句话）；

比较两种提交对于轮播图片的影响。

# 四、代码及运行结果

## 4.1 图片轮播页面截图



## 4.2 图片轮播HTML页面代码

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>ajax图片轮播</title>

<link rel="stylesheet" href="../css/ajax\_index.css" />

</head>

<body>

<div id="all">

<div id="left">

<header>

图片轮播

</header>

<section>

<img id="imgid" src="../img/ajax/1.jpg" />

</section>

<footer>

<label id="num1">1</label>

<label id="num2">2</label>

<label id="num3">3</label>

<label id="num4">4</label>

<label id="num5">5</label>

<label id="num6">6</label>

<label id="num7">7</label>

</footer>

</div>

<div id="right">

<div id="right\_top">

<form method="post">

<ul>

<li>

<label>用户名：</label>

<span><input type="text" /></span>

</li>

<li>

<label>密码：</label>

<span><input type="text" /></span>

</li>

<li>

<input class="btn" type="button" value="普通提交"/>

<input class="btn" type="reset" value="重新填写"/>

</li>

</ul>

</form>

</div>

<div id="right\_bottom">

<ul>

<li>

<label>用户名：</label>

<span><input type="text" /></span>

</li>

<li>

<label>用户名：</label>

<span><input type="text" /></span>

</li>

<li>

<input class="btn" type="button" value="ajax提交"/>

<input class="btn" type="reset" value="重新填写"/>

</li>

</ul>

</div>

</div>

</div>

<script type="text/javascript" src="../js/ajax/ajax\_index.js" ></script>

</body>

</html>

## 4.4 图片轮播CSS代码

body{

*margin*: 0;

}

#all{

*width*: 1200px;

*height*: 660px;

*border*:1px solid black;

*margin*: auto;

/\*background: #000000;\*/

}

#all #left{

*border*: 1px solid green;

*width*: 49%;

*height*: 660px;

*float*: *left*;

}

#all #left header{

*height*: 80px;

*background*: aqua;

*text-align*: center;

*line-height*: 80px;

*font-family*:"bookshelf symbol 7";

*font-size*: 40px;

}

#all #left section{

*height*: 520px;

}

#all #left section img{

*width*: 100%;

*height*: 100%;

}

#all #left footer{

*height*: 60px;

*background*: orange;

*text-align*: center;

*font-size*: 20px;

*line-height*: 60px;

}

#all #left footer label{

*border*: 1px solid yellow;

*margin*: 10px;

*padding*: 10px 20px 10px 20px;

*background*: #DA70D6;

}

#all #right{

*border*: 1px solid green;

*background*: #DA70D6;

*width*: 49%;

*height*: 660px;

*float*: *right*;

}

#all #right ul{

*margin*: 0;

/\* background: aliceblue;\*/

*list-style*: none;

*padding*:60px 20px 20px 20px;

*margin-left*: 120px;

}

#all #right ul li{

*border*: 1px solid black;

*width*: 70%;

*height*: 60px;

*text-align*: center;

*line-height*: 60px;

}

#all #right ul li .btn{

*margin*: 0px 20px 20px 20px ;

}

#all #right #right\_top{

*background*: yellow;

*height*: 330px;

}

#all #right #right\_bottom{

*background*: blue;

*height*: 330px;

}

## 4.5 图片轮播js代码

*var* i=1;

go2();

*function* go2()

{

*var* imgid= *document*.getElementById("imgid");

imgid.src="../img/ajax/"+i+".jpg";

for(*var* j=1;j<8;j++)

{

*var* numid0=*document*.getElementById("num"+j) ;

numid0.className="one";

}

*var* numid=*document*.getElementById("num"+i) ;

numid.className="two";

i++;

if (i==8)

{

i=1;

}

setTimeout("go2()",1000);

}

# 五、思考题

1、ajax的onreadystatechange事件中readyState有几种状态？分别是什么含义？

2、ajax的定义？

3、ajax的主要特点是什么？



福建师范大学协和学院《人机交互》实验报告

学号：\_\_\_\_\_\_\_\_ 姓名：\_\_\_\_\_\_\_\_\_\_ 实验日期：\_2019.5.16 成绩：\_\_\_\_\_\_\_

实验六 交互系统—Ueditor图片上传

# 一、实验目的

UEditor是由百度web前端研发部开发所见即所得富文本web编辑器，具有轻量，可定制，注重用户体验等特点，开源基于MIT协议，允许自由使用和修改代码，使用UEditor开发web前端界面，能够同时添加图片和文字。通过本次实验，达到以下目的：

（1）了解Ueditor包含哪些功能；

（2）掌握mui框架的ajax方法；

（3）掌握百度Ueditor控件的配置方法；

（4）使用Ueditor同时上传文字+图片。

二、实验环境（主要仪器设备和软件）

1、操作系统Windows 7

2、HBuilder.

3、Google Chrome

4、tomcat

5、eclipse

6、jdk

# 三、实验内容

设计一个产品添加的前端界面，何种产品可根据自己的项目需求自由确定。要求添加产品名称，产品特性，产品介绍等，其中添加产品介绍使用UEditor编辑器进行图文编辑。

**注：请大家自由选择自己的业务，本题目以新闻为例，大家选择不能是新闻。**

（1）Hbuilder端设计新闻添加界面；

（2）在html页面添加ueditor控件；

（3）在eclipse中创建对应的jsp页面，css页面；

（4）创建数据库；

（5）创建javabean；

（6）创建Servlet类；

（7）web.xml文件中创建其servlet配置；

（8）js中创建news文件夹，其下常见addnews.js文件；

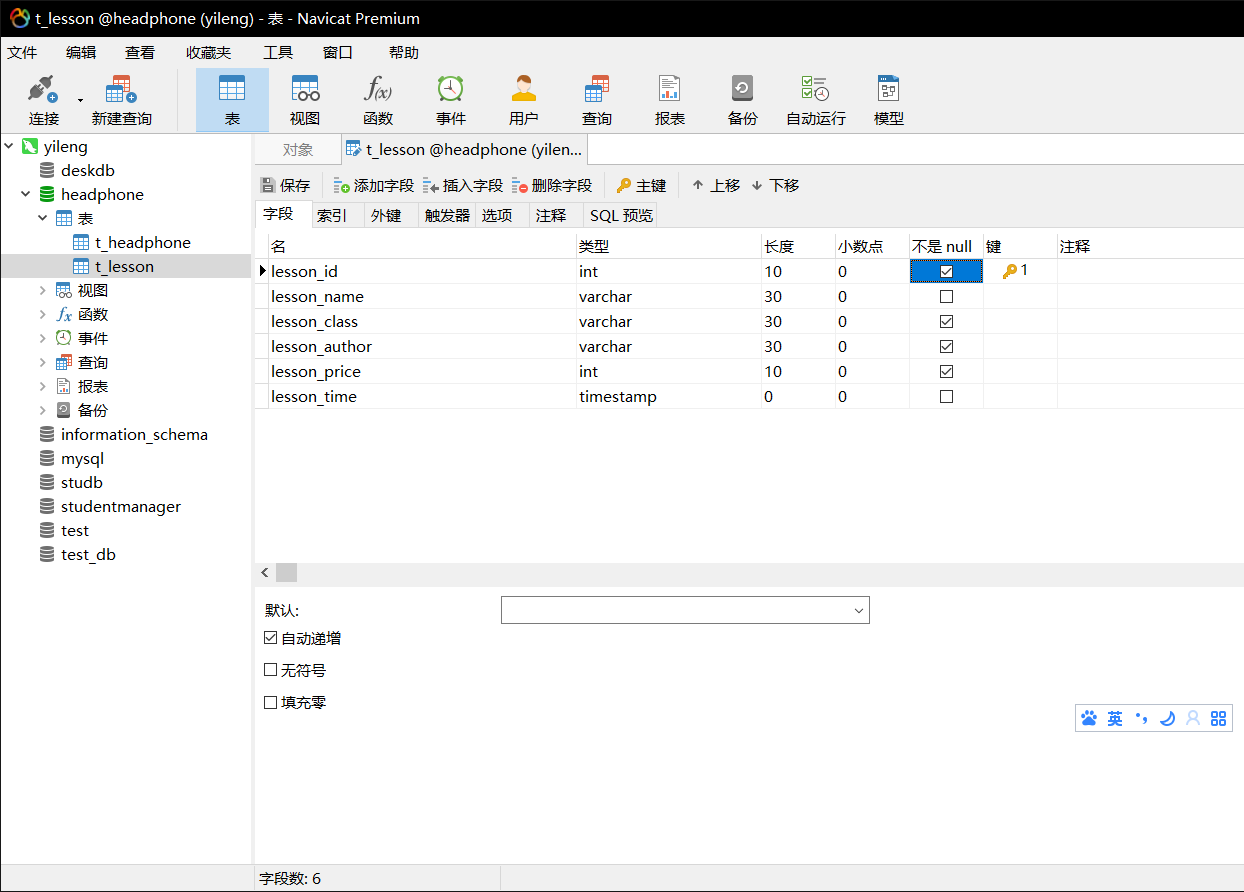
（9）编写 add\_news()方法；

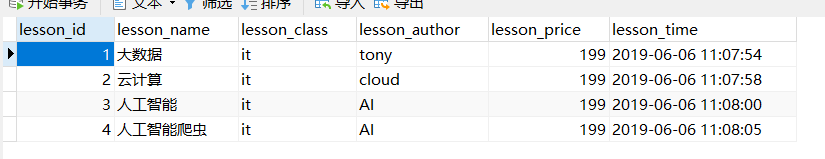
（10）让图片在编辑框中显示出来。

（11）使用mui的ajax，将内容添加到数据库中；

# 四、代码及运行结果

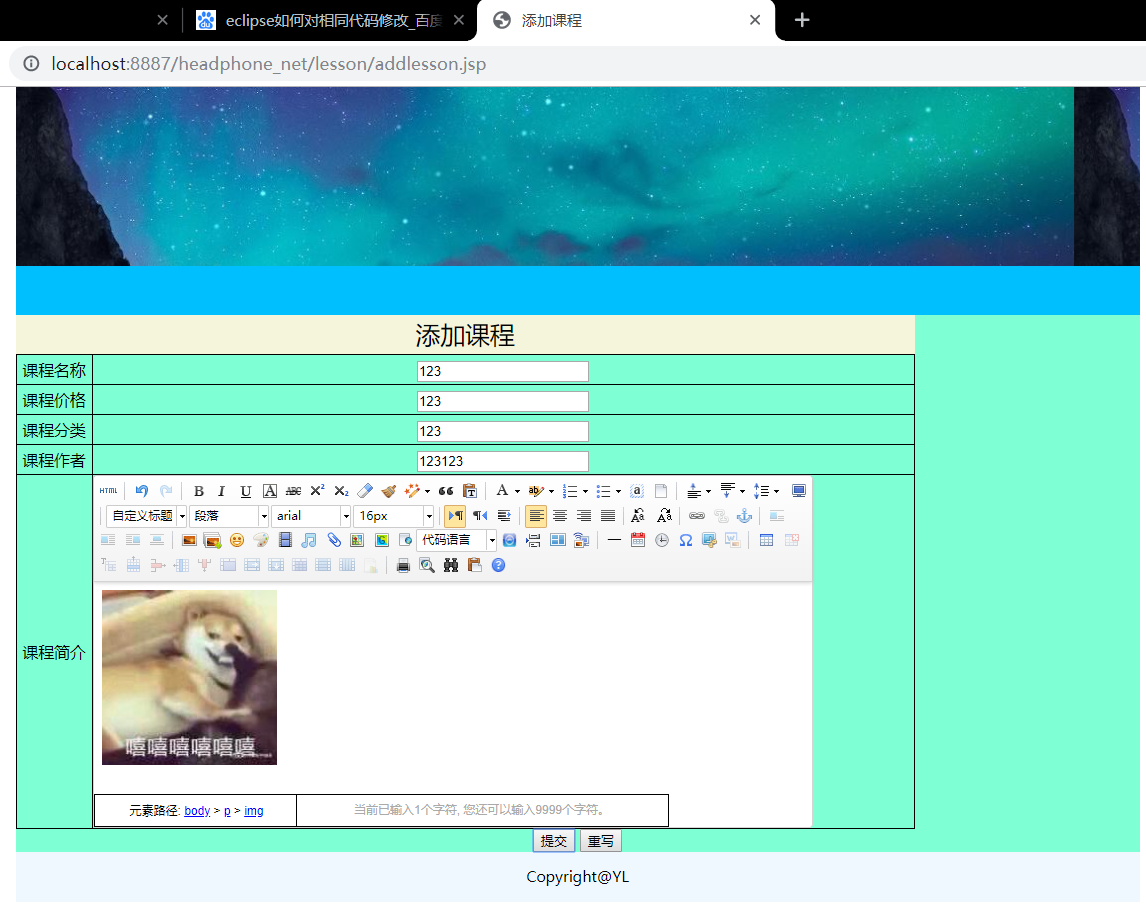
## 4.1 数据库和数据表截图





## 4.2 图片上传页面

### 4.2.1 页面效果截图



### 4.2.2 jsp代码

<%@page import="org.apache.jasper.tagplugins.jstl.core.If"%>

<%@page import="com.head\_phone.beans.\*"%>

<%@page import="java.util.\*"%>

<%@ page language="java" contentType="text/html; charset=utf-8"

pageEncoding="utf-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>添加课程</title>

<link rel="stylesheet" type="text/css" href="../css/lesson/addlesson.css"/>

</head>

<body>

<div id="all">

<div id="top">

</div>

<div id="nav">

</div>

<div id="main">

<table>

<caption>添加课程</caption>

<tr>

<td>课程名称</td>

<td><input type="text" name="lesson\_name" id="lesson\_name" /></td>

</tr>

<tr>

<td>课程价格</td>

<td><input type="text" name="lesson\_price" id="lesson\_price" /></td>

</tr>

<tr>

<td>课程分类</td>

<td><input type="text" name="lesson\_class" id="lesson\_class" /></td>

</tr>

<tr>

<td>课程作者</td>

<td><input type="text" name="lesson\_author" id="lesson\_author" /></td>

</tr>

<tr>

<td>课程简介</td>

<script type="text/plain" id="lessonsEditor">

</script>

</tr>

</table>

<div id="button">

<input class="btstyle" type="button" onclick="add\_lesson()" value="提交" />

<input class="btstyle" type="reset" value="重写" />

</div>

</div>

<div id="foot">

Copyright@YL

</div>

</div>

<script src="../js/mui.js"type="text/javascript" charset="UTF-8"></script>

<script src="../js/mui.min.js"type="text/javascript" charset="UTF-8"></script>

<script src="../utf8-jsp/ueditor.config.js" type="text/javascript" charset="UTF-8"></script>

<script src="../utf8-jsp/ueditor.all.js" type="text/javascript" charset="UTF-8"></script>

<script src="../js/lesson/addlesson.js" type="text/javascript" charset="UTF-8"></script>

</body>

</html>

### 4.2.3 CSS代码

body{

margin: *0*;

}

*#all* *#main*{

height: *auto*;

}

*#all*{

width: *80%*;

height: *600px*;

/\*border: 1px solid black ;\*/

margin: *auto*;

}

*#all* *#top*{

height: *180px*;

background-image: *url(../../img/headphone/topbg.JPG)*;

}

*#nav*{

height: *50px*;

background: *deepskyblue*;

}

*#main*{

height: *400px*;

background-color: *aquamarine*

}

*#main* table{

border: *1px* *solid* *black*;

width: *80%*;

border-collapse: *collapse*;

}

*#main* table caption{

background: *beige*;

line-height: *40px*;

font-size: *25px*;

font-family: *"寰蒋闆呴粦"*;

}

*#main* table td{

border:*1px* *solid* *black*;

line-height: *30px*;

text-align: *center*;

}

*#button*{

text-align:*center*;

}

*#foot*{

height: *50px*;

background: *aliceblue*;

font-size: *15px*;

font-family: *"寰蒋闆呴粦"*;

line-height: *50px*;

text-align:*center*;

}

*#button1*{

padding: *30px*;

height: *80px*;

text-align: *center*;

}

*#button1* button{

height: *30px*;

width: *80px*;

}

*#button2*{

height: *50px*;

text-align: *center*;

}

*#button2* button{

height: *30px*;

width: *80px*;

}

### 4.2.4 js代码

**var** ue=UE.getEditor('lessonsEditor');

**function** add\_lesson(){

**var** lesson\_name=document.getElementById("lesson\_name").value;

**var** lesson\_class=document.getElementById("lesson\_class").value;

**var** lesson\_author=document.getElementById("lesson\_author").value;

**var** lesson\_price=document.getElementById("lesson\_price").value;

**var** lesson\_text=ue.getContent();

**var** lessonJsonObj={

"lesson\_name":lesson\_name,

"lesson\_class":lesson\_class,

"lesson\_author":lesson\_author,

"lesson\_price":lesson\_price

};

console.log("lessonJsonObj="+lessonJsonObj);

**var** lessonJsonStr=JSON.stringify(lessonJsonObj)

console.log("lessonJsonStr="+lessonJsonStr);

mui.ajax('http://localhost:8887/headphone\_net/servlet/lessonservlet?para=add\_lesson\_pc',{

data:lessonJsonObj,

dataType:'json',//服务器返回json格式数据

type:'post',//HTTP请求类型

timeout:0,//超时时间设置为5秒；

headers:{'Content-Type':'application/json'},

success:**function**(data){

//服务器返回响应，根据响应结果，分析是否登录成功；

**var** value=data.msg;

alert(value);

},

error:**function**(xhr,type,errorThrown){

//异常处理；

console.log(type);

}

});

}

## 4.3 dao函数

package com.headphone\_net.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Timestamp;

import java.util.ArrayList;

import com.head\_phone.beans.lesson;

import com.headphone\_src.utils.DBUtil;

public class LessondaoImpl {

public boolean modifylesson(lesson lesson)

{

boolean flag=false;

DBUtil dbUtil=new DBUtil();

Connection conn= (Connection) dbUtil.getConn();

try {

String sql="update t\_lesson set lesson\_name=?,lesson\_price=?,lesson\_class=? where phone\_id=?";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setString(1,lesson.getLesson\_name());

pstmt.setString(2,lesson.getLesson\_class());

pstmt.setString(3,lesson.getLesson\_author());

pstmt.setInt(4,lesson.getLesson\_price());

pstmt.setTimestamp(5,lesson.getLesson\_time() );

int result=pstmt.executeUpdate();

if (1==result) {

flag=true;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

try {

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return flag;

}

public boolean insertlesson(lesson lesson)

{

boolean flag=false;

DBUtil dbUtil=new DBUtil();

Connection conn= (Connection) dbUtil.getConn();

try {

String sql="insert into t\_lesson(lesson\_name,lesson\_price,lesson\_class,lesson\_author,lesson\_time) values(?,?,?,?,?)";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setString(1,lesson.getLesson\_name());

pstmt.setInt(2,lesson.getLesson\_price());

pstmt.setString(3,lesson.getLesson\_class());

pstmt.setString(4,lesson.getLesson\_author());

pstmt.setTimestamp(5,lesson.getLesson\_time() );

int result=pstmt.executeUpdate();

if (1==result) {

flag=true;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

try {

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return flag;

}

//按照id进行查询。

public lesson searchlessonById(int lesson\_id\_para) {

lesson lesson=new lesson();

DBUtil dbUtil=new DBUtil();

Connection conn= (Connection) dbUtil.getConn();

ResultSet rs=null;

try {

String sql="select \* from t\_lesson where lesson\_id=?";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setInt(1, lesson\_id\_para);

rs=pstmt.executeQuery();

while (rs.next()) {

int lesson\_id=rs.getInt("lesson\_id");

String lesson\_name=rs.getString("lesson\_name");

float lesson\_price=rs.getFloat("lesson\_price");

String lesson\_class=rs.getString("lesson\_class");

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

try {

rs.close();

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return lesson;

}

public ArrayList<lesson> searchLessonByName(String lesson\_name\_para)

{

ArrayList<lesson> lessonlist=new ArrayList<lesson>();

DBUtil dBUtil=new DBUtil();

Connection conn= dBUtil.getConn();

ResultSet rs=null;

PreparedStatement psmt=null;

String sql="";

try {

if(lesson\_name\_para.equals("")||lesson\_name\_para==null){

sql="select \* from t\_lesson";

psmt=conn.prepareStatement(sql);

}else {

sql="select \* from t\_lesson where lesson\_name=?";

psmt=conn.prepareStatement(sql);

psmt.setString(1, lesson\_name\_para);

}

rs=psmt.executeQuery();

while(rs.next())

{

int lesson\_id=rs.getInt("lesson\_id");

String lesson\_name=rs.getString("lesson\_name");

String lesson\_class=rs.getString("lesson\_class");

String lesson\_author=rs.getString("lesson\_author");

int lesson\_price=rs.getInt("lesson\_price");

Timestamp lesson\_time=rs.getTimestamp("lesson\_time");

lesson lesson=new lesson();

lesson.setLesson\_id(lesson\_id);

lesson.setLesson\_name(lesson\_name);

lesson.setLesson\_class(lesson\_class);

lesson.setLesson\_author(lesson\_author);

lesson.setLesson\_price(lesson\_price);

lesson.setLesson\_time(lesson\_time);

lessonlist.add(lesson);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

try {

rs.close();

conn.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

return lessonlist;

}

public boolean deleteByID(int lesson\_id\_para) {

boolean flag=false;

DBUtil dbUtil=new DBUtil();

Connection conn=dbUtil.getConn();

String sql="delete from t\_lesson where lesson\_id=?";

try {

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setInt(1,lesson\_id\_para);

int result=pstmt.executeUpdate();

if (1==result) {

flag=true;

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

try {

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return flag;

}

// public static void main(String[] args) {

// lessondaoImpl lessondaoImpl=new lessondaoImpl();

// lesson lesson=new lesson();

// lesson.setlesson\_name("森海塞尔");

// lesson.setlesson\_price(799);

// lesson.setlesson\_class("头戴式");

// boolean flag=lessondaoImpl.insertlesson(lesson);

// System.out.println(flag);

// }

}

## 4.4 servlet函数

package com.headphone\_net.servlet;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Timestamp;

import java.util.ArrayList;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.google.gson.Gson;

import com.head\_phone.beans.lesson;

import com.headphone\_net.dao.LessondaoImpl;

public class LessonServlet extends HttpServlet{

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

request.setCharacterEncoding("utf-8");

response.setContentType("text/html;charset=utf-8");

doPost(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

request.setCharacterEncoding("utf-8");

response.setContentType("text/html;charset=utf-8");

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

if(para.equals("add\_lesson\_pc"))

{

add\_lesson\_pc(request,response);

System.out.println("jsonStr");

}else if(para.equals("search\_lesson\_pc"))

{

search\_lesson\_pc(request,response);

System.out.println("6666");

}

}

private void search\_lesson\_pc(HttpServletRequest request, HttpServletResponse response) {

// TODO Auto-generated method stub

String jsonStr=readJSONObj(request);

System.out.println("jsonStr"+jsonStr);

Gson gson=new Gson();

lesson lesson=gson.fromJson(jsonStr, lesson.class);

System.out.println("Lesson"+lesson);

String lesson\_name=lesson.getLesson\_name();

LessondaoImpl lessondaoImpl=new LessondaoImpl();

ArrayList<lesson> lessonlist=lessondaoImpl.searchLessonByName(lesson\_name);

String json\_array=gson.toJson(lessonlist);

System.out.println(json\_array);

try {

PrintWriter out = response.getWriter();

out.println(json\_array);

out.flush();

out.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

private void add\_lesson\_pc(HttpServletRequest request, HttpServletResponse response) {

// TODO Auto-generated method stub

String jsonStr=readJSONObj(request);

System.out.println("jsonStr"+jsonStr);

Gson gson=new Gson();

lesson lesson=gson.fromJson(jsonStr, lesson.class);

System.out.println("lesson"+lesson);

Timestamp lesson\_time=new Timestamp(System.currentTimeMillis());

System.out.println("lesson\_time"+lesson\_time);

lesson.setLesson\_time(lesson\_time);

LessondaoImpl lessondaoimp=new LessondaoImpl();

boolean flag=lessondaoimp.insertlesson(lesson);

String serverJsonStr ="{\"msg\":\"add success!\"}";

if(!flag)

{

serverJsonStr ="{\"msg\":\"add failure!\"}";

}

try {

PrintWriter out=response.getWriter();

out.println(serverJsonStr);

out.flush();

out.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public String readJSONObj(HttpServletRequest req){

String jsonStr="";

StringBuffer jsonStrBuffer=new StringBuffer();

try {

BufferedReader reader=req.getReader();

String line=reader.readLine();

while(line!=null)

{

jsonStrBuffer.append(line);

line=reader.readLine();

}

jsonStr=jsonStrBuffer.toString();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return jsonStr;

}

}

## 4.5 web.xml中该模块serlvet的配置。

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://java.sun.com/xml/ns/javaee"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"* id=*"WebApp\_ID"* version=*"3.0"*>

<display-name>headphone\_net</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>HeadphoneServlet</servlet-name>

<servlet-class>com.headphone\_net.servlet.HeadphoneSerlvet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>HeadphoneServlet</servlet-name>

<url-pattern>/servlet/headphoneservlet</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>AjaxServlet</servlet-name>

<servlet-class>com.headphone\_net.servlet.AjaxServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>AjaxServlet</servlet-name>

<url-pattern>/servlet/ajaxservlet</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>LessonServlet</servlet-name>

<servlet-class>com.headphone\_net.servlet.LessonServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>LessonServlet</servlet-name>

<url-pattern>/servlet/lessonservlet</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>CodeServlet</servlet-name>

<servlet-class>com.headphone\_net.servlet.CodeServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>CodeServlet</servlet-name>

<url-pattern>/servlet/codeservlet</url-pattern>

</servlet-mapping>

</web-app>

## 4.6该模块的javabean代码

**package** com.head\_phone.beans;

**import** java.sql.Timestamp;

**public** **class** lesson {

**private** **int** lesson\_id;

**private** String lesson\_name;

**private** String lesson\_class;

**private** String lesson\_author;

**private** **int** lesson\_price;

**private** Timestamp lesson\_time;

**public** **int** getLesson\_id() {

**return** lesson\_id;

}

**public** **void** setLesson\_id(**int** lesson\_id) {

**this**.lesson\_id = lesson\_id;

}

**public** String getLesson\_name() {

**return** lesson\_name;

}

**public** **void** setLesson\_name(String lesson\_name) {

**this**.lesson\_name = lesson\_name;

}

**public** String getLesson\_class() {

**return** lesson\_class;

}

**public** **void** setLesson\_class(String lesson\_class) {

**this**.lesson\_class = lesson\_class;

}

**public** String getLesson\_author() {

**return** lesson\_author;

}

**public** **void** setLesson\_author(String lesson\_author) {

**this**.lesson\_author = lesson\_author;

}

**public** **int** getLesson\_price() {

**return** lesson\_price;

}

**public** **void** setLesson\_price(**int** lesson\_price) {

**this**.lesson\_price = lesson\_price;

}

**public** Timestamp getLesson\_time() {

**return** lesson\_time;

}

**public** **void** setLesson\_time(Timestamp lesson\_time) {

**this**.lesson\_time = lesson\_time;

}

@Override

**public** String toString() {

**return** "lesson [lesson\_id=" + lesson\_id + ", lesson\_name=" + lesson\_name + ", lesson\_class=" + lesson\_class

+ ", lesson\_author=" + lesson\_author + ", lesson\_price=" + lesson\_price + ", lesson\_time=" + lesson\_time

+ "]";

}

}

# 五、思考题

1、Ueditor是哪个公司开发的？都包含哪些功能？

2、Ueditor在jsp页面添加的代码如何表示？

3、如何在js中创建Ueditor对象？



福

建师范大学协和学院《人机交互》实验报告

学号：\_\_\_\_\_\_\_\_ 姓名：\_\_\_\_\_\_\_\_\_\_ 实验日期：\_2019.5.30 成绩：\_\_\_\_\_\_\_

实验七 交互系统—VUE.js查询新闻-PC

# 一、实验目的

Vue.js是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与[现代化的工具链](https://cn.vuejs.org/v2/guide/single-file-components.html)以及各种[支持类库](https://github.com/vuejs/awesome-vue#libraries--plugins)结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。通过本次实验，达到以下目的：

1. 掌握vue的模板语法；
2. 掌握vue的条件语句、循环语句等语句；
3. 掌握vue的事件处理器；
4. 掌握vue的样式绑定、表单路由等。

# 二、实验环境（主要仪器设备和软件）

1、操作系统Windows 7

2、HBuilder.

3、Google Chrome

4、tomcat

5、eclipse

6、jdk

# 三、实验内容

在实验六的基础上，引入vue框架，结合javaEE，实现带图片的表格的数据的查询。

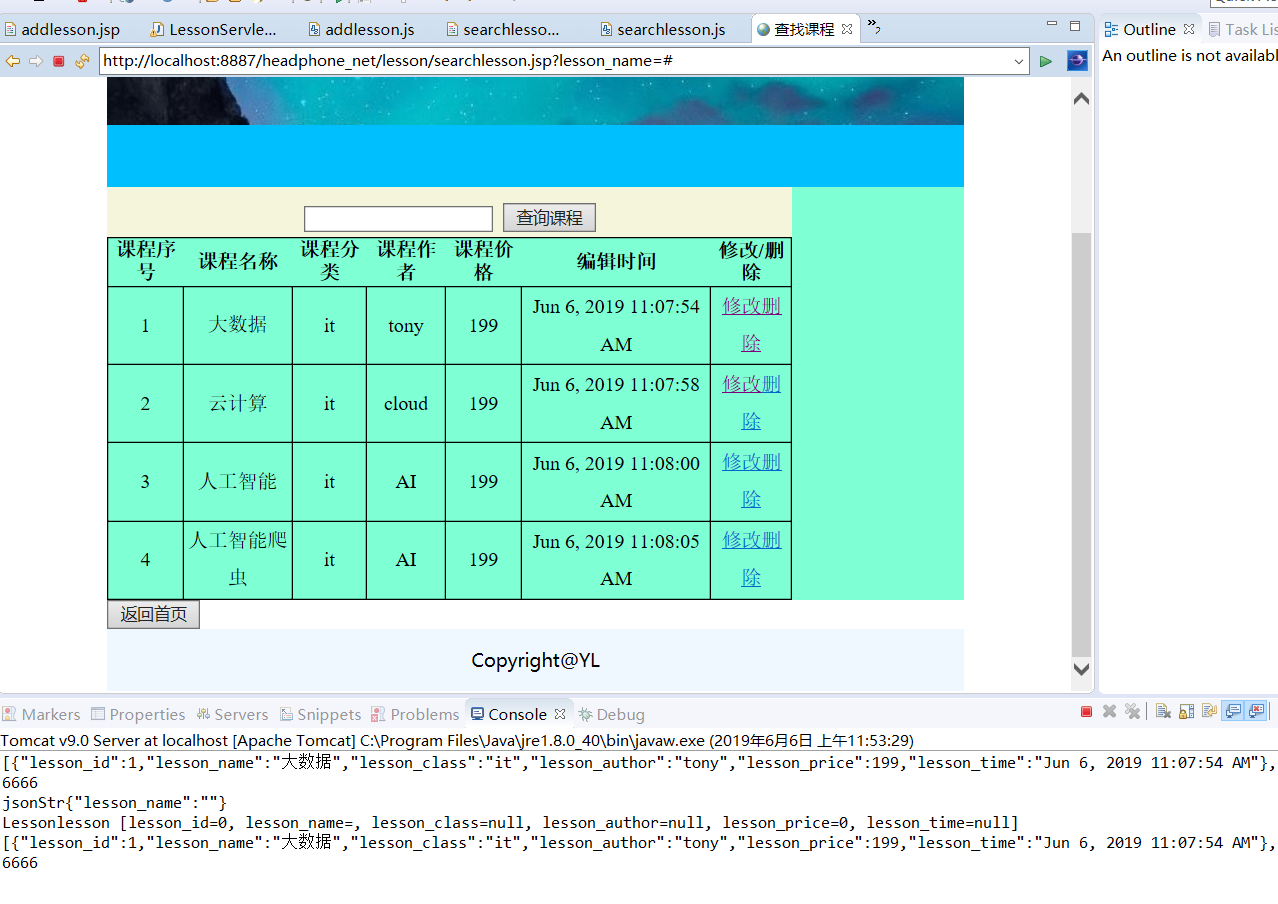
1. PC端的新闻查询页面；
2. dao的新闻查询代码；
3. servlet的新闻查询代码；

# 四、代码及运行结果

## 4.1 图片表格查询页面

### 4.1.1 页面效果截图





### 4.1.2 jsp代码

<%@page import="org.apache.jasper.tagplugins.jstl.core.If"%>

<%@page import="com.head\_phone.beans.\*"%>

<%@page import="java.util.\*"%>

<%@ page language="java" contentType="text/html; charset=utf-8"

pageEncoding="utf-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>查找课程</title>

<link rel="stylesheet" type="text/css" href="../css/lesson/addlesson.css"/>

</head>

<body>

<div id="all">

<div id="top">

</div>

<div id="nav">

</div>

<div id="main">

<form >

<table id="lessontable">

<caption>

<input type="text" name="lesson\_name" id="lesson\_name" />

<input type="submit" value="查询课程" onclick="search\_lesson()" />

</caption>

<tr>

<th>课程序号</th>

<th>课程名称</th>

<th>课程分类</th>

<th>课程作者</th>

<th>课程价格</th>

<th>编辑时间</th>

<th>修改/删除</th>

</tr>

</table>

</form>

</div>

<a href="http://localhost:8887/headphone\_net/headphone/index.jsp"></a>

<button>返回首页</button>

<div id="foot">

Copyright@YL

</div>

</div>

<script src="../js/mui.js" charset="utf-8" type="text/javascript"></script>

<script src="../js/mui.min.js"type="text/javascript" charset="UTF-8"></script>

<script src="../js/lesson/searchlesson.js" charset="utf-8" type="text/javascript"></script>

</body>

</html>

### 4.1.3 CSS代码

body{

margin: *0*;

}

*#all* *#main*{

height: *auto*;

}

*#all*{

width: *80%*;

height: *600px*;

/\*border: 1px solid black ;\*/

margin: *auto*;

}

*#all* *#top*{

height: *180px*;

background-image: *url(../../img/headphone/topbg.JPG)*;

}

*#nav*{

height: *50px*;

background: *deepskyblue*;

}

*#main*{

height: *400px*;

background-color: *aquamarine*

}

*#main* table{

border: *1px* *solid* *black*;

width: *80%*;

border-collapse: *collapse*;

}

*#main* table caption{

background: *beige*;

line-height: *40px*;

font-size: *25px*;

font-family: *"寰蒋闆呴粦"*;

}

*#main* table td{

border:*1px* *solid* *black*;

line-height: *30px*;

text-align: *center*;

}

*#button*{

text-align:*center*;

}

*#foot*{

height: *50px*;

background: *aliceblue*;

font-size: *15px*;

font-family: *"寰蒋闆呴粦"*;

line-height: *50px*;

text-align:*center*;

}

*#button1*{

padding: *30px*;

height: *80px*;

text-align: *center*;

}

*#button1* button{

height: *30px*;

width: *80px*;

}

*#button2*{

height: *50px*;

text-align: *center*;

}

*#button2* button{

height: *30px*;

width: *80px*;

}

### 4.1.4 js代码

**function** search\_lesson()

{

**var** lesson\_name=document.getElementById("lesson\_name").value;

console.log("lesson\_name="+lesson\_name);

**var** lessonJsonObj={"lesson\_name":lesson\_name};

mui.ajax('http://localhost:8887/headphone\_net/servlet/lessonservlet?para=search\_lesson\_pc',{

data:lessonJsonObj,

dataType:'json',//服务器返回json格式数据

type:'post',//HTTP请求类型

timeout:0,//超时时间设置为10秒；

headers:{'Content-Type':'application/json'},

success:**function**(data){

//服务器返回响应，根据响应结果，分析是否登录成功

display(data);

},

error:**function**(xhr,type,errorThrown){

//异常处理；

console.log(type);

}

});

}

**function** display(json\_array){

**var** lessontable=document.getElementById("lessontable");

**for**(**var** i=0;i<json\_array.length;i++)

{

**var** json\_obj=json\_array[i];

**var** json\_str=JSON.stringify(json\_obj);

console.log("jsonstr="+json\_str);

**var** lesson\_id=json\_obj.lesson\_id;

**var** lesson\_name=json\_obj.lesson\_name;

**var** lesson\_class=json\_obj.lesson\_class;

**var** lesson\_author=json\_obj.lesson\_author;

**var** lesson\_price=json\_obj.lesson\_price;

**var** lesson\_time=json\_obj.lesson\_time;

**var** tr=document.createElement("tr");

**var** td1=document.createElement("td");

**var** td2=document.createElement("td");

**var** td3=document.createElement("td");

**var** td4=document.createElement("td");

**var** td5=document.createElement("td");

**var** td6=document.createElement("td");

**var** td7=document.createElement("td");

td1.innerHTML=lesson\_id;

td2.innerHTML=lesson\_name;

td3.innerHTML=lesson\_class;

td4.innerHTML=lesson\_author;

td5.innerHTML=lesson\_price;

td6.innerHTML=lesson\_time;

td7.innerHTML='<a href="#">修改</a>'+'<a href="#">删除</a>'

tr.appendChild(td1);

tr.appendChild(td2);

tr.appendChild(td3);

tr.appendChild(td4);

tr.appendChild(td5);

tr.appendChild(td6);

tr.appendChild(td7);

lessontable.appendChild(tr);

}

}

## 4.2查询的 dao函数

**public** lesson searchlessonById(**int** lesson\_id\_para) {

lesson lesson=**new** lesson();

DBUtil dbUtil=**new** DBUtil();

Connection conn= (Connection) dbUtil.getConn();

ResultSet rs=**null**;

**try** {

String sql="select \* from t\_lesson where lesson\_id=?";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt.setInt(1, lesson\_id\_para);

rs=pstmt.executeQuery();

**while** (rs.next()) {

**int** lesson\_id=rs.getInt("lesson\_id");

String lesson\_name=rs.getString("lesson\_name");

**float** lesson\_price=rs.getFloat("lesson\_price");

String lesson\_class=rs.getString("lesson\_class");

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}**finally** {

**try** {

rs.close();

conn.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**return** lesson;

}

**public** ArrayList<lesson> searchLessonByName(String lesson\_name\_para)

{

ArrayList<lesson> lessonlist=**new** ArrayList<lesson>();

DBUtil dBUtil=**new** DBUtil();

Connection conn= dBUtil.getConn();

ResultSet rs=**null**;

PreparedStatement psmt=**null**;

String sql="";

**try** {

**if**(lesson\_name\_para.equals("")||lesson\_name\_para==**null**){

sql="select \* from t\_lesson";

psmt=conn.prepareStatement(sql);

}**else** {

sql="select \* from t\_lesson where lesson\_name=?";

psmt=conn.prepareStatement(sql);

psmt.setString(1, lesson\_name\_para);

}

rs=psmt.executeQuery();

**while**(rs.next())

{

**int** lesson\_id=rs.getInt("lesson\_id");

String lesson\_name=rs.getString("lesson\_name");

String lesson\_class=rs.getString("lesson\_class");

String lesson\_author=rs.getString("lesson\_author");

**int** lesson\_price=rs.getInt("lesson\_price");

Timestamp lesson\_time=rs.getTimestamp("lesson\_time");

lesson lesson=**new** lesson();

lesson.setLesson\_id(lesson\_id);

lesson.setLesson\_name(lesson\_name);

lesson.setLesson\_class(lesson\_class);

lesson.setLesson\_author(lesson\_author);

lesson.setLesson\_price(lesson\_price);

lesson.setLesson\_time(lesson\_time);

lessonlist.add(lesson);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}**finally** {

**try** {

rs.close();

conn.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**return** lessonlist;

}

## 4.3 查询的servlet函数

package com.headphone\_net.servlet;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Timestamp;

import java.util.ArrayList;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.google.gson.Gson;

import com.head\_phone.beans.lesson;

import com.headphone\_net.dao.LessondaoImpl;

public class LessonServlet extends HttpServlet{

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

request.setCharacterEncoding("utf-8");

response.setContentType("text/html;charset=utf-8");

doPost(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

request.setCharacterEncoding("utf-8");

response.setContentType("text/html;charset=utf-8");

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

if(para.equals("add\_lesson\_pc"))

{

add\_lesson\_pc(request,response);

System.out.println("jsonStr");

}else if(para.equals("search\_lesson\_pc"))

{

search\_lesson\_pc(request,response);

System.out.println("6666");

}

}

private void search\_lesson\_pc(HttpServletRequest request, HttpServletResponse response) {

// TODO Auto-generated method stub

String jsonStr=readJSONObj(request);

System.out.println("jsonStr"+jsonStr);

Gson gson=new Gson();

lesson lesson=gson.fromJson(jsonStr, lesson.class);

System.out.println("Lesson"+lesson);

String lesson\_name=lesson.getLesson\_name();

LessondaoImpl lessondaoImpl=new LessondaoImpl();

ArrayList<lesson> lessonlist=lessondaoImpl.searchLessonByName(lesson\_name);

String json\_array=gson.toJson(lessonlist);

System.out.println(json\_array);

try {

PrintWriter out = response.getWriter();

out.println(json\_array);

out.flush();

out.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

private void add\_lesson\_pc(HttpServletRequest request, HttpServletResponse response) {

// TODO Auto-generated method stub

String jsonStr=readJSONObj(request);

System.out.println("jsonStr"+jsonStr);

Gson gson=new Gson();

lesson lesson=gson.fromJson(jsonStr, lesson.class);

System.out.println("lesson"+lesson);

Timestamp lesson\_time=new Timestamp(System.currentTimeMillis());

System.out.println("lesson\_time"+lesson\_time);

lesson.setLesson\_time(lesson\_time);

LessondaoImpl lessondaoimp=new LessondaoImpl();

boolean flag=lessondaoimp.insertlesson(lesson);

String serverJsonStr ="{\"msg\":\"add success!\"}";

if(!flag)

{

serverJsonStr ="{\"msg\":\"add failure!\"}";

}

try {

PrintWriter out=response.getWriter();

out.println(serverJsonStr);

out.flush();

out.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public String readJSONObj(HttpServletRequest req){

String jsonStr="";

StringBuffer jsonStrBuffer=new StringBuffer();

try {

BufferedReader reader=req.getReader();

String line=reader.readLine();

while(line!=null)

{

jsonStrBuffer.append(line);

line=reader.readLine();

}

jsonStr=jsonStrBuffer.toString();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return jsonStr;

}

}

# 五、思考题

1、简单介绍什么是vue?

2、MVC框架与MVVM的区别？

3、简单介绍什么是React.js框架？

福建师范大学协和学院《人机交互》实验报告

学号：\_\_\_\_\_\_\_\_ 姓名：\_\_\_\_\_\_\_\_\_\_ 实验日期：\_2019.6.6 成绩：\_\_\_\_\_\_\_

实验八 交互系统—百度Echart（PC+APP）

# 一、实验目的

ECharts商业级数据图表，它是一个纯JavaScript的图标库，兼容绝大部分的浏览器，底层依赖轻量级的canvas类库ZRender，提供直观，生动，可交互，可高度个性化定制的数据可视化图表。通过本次实验，完成以下目的：

（1）掌握Echarts折线图、柱状图、饼图的前端开发；

（2）掌握Echart在PC系统的实现；

（3）掌握Echart在APP系统的实现。

二、实验环境（主要仪器设备和软件）

1、操作系统Windows 7

2、HBuilder.

3、Google Chrome

4、tomcat

5、eclipse

6、jdk

# 三、实验内容

要求：图表数据必须从服务器端获得（手动输入到数据库）。

（1）、在数据库中创建表：销量表。

（2）、创建产品销量的javabean。

（3）、编写查询销量的dao代码（phone模块）。

（4）、在web.xml中编写product的servlet

（5）、创建ProductServlet,并编写相应的代码；

（6）、在浏览器端写ajax查询的js代码

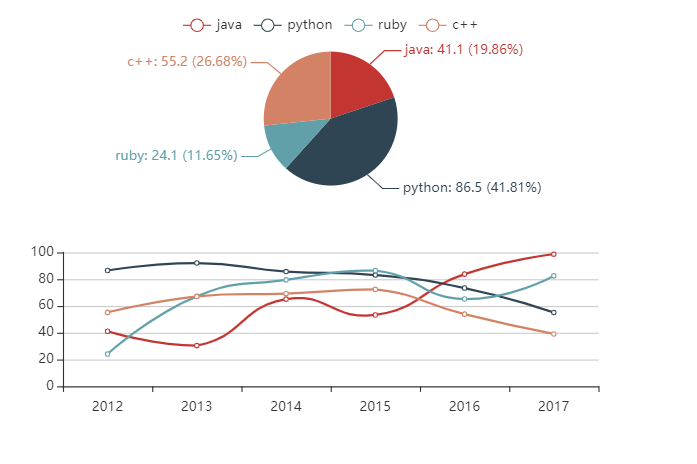
（7）、调试图表在页面进行显示

（8）、简单做一个移动端的页面，使用图标在移动端页面进行显示。

# 四、代码及运行结果

## 4.1 PC端页面效果

### 4.1.1 页面效果截图



### 4.1.2 jsp代码

<%@ page language="java" contentType="text/html; charset=utf-8"

pageEncoding="utf-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>ECharts</title>

<script src="../js/mui.js"type="text/javascript" charset="UTF-8"></script>

<script src="../js/mui.min.js"type="text/javascript" charset="UTF-8"></script>

<script type="text/javascript" src="../js/echarts.js" ></script>

<script type="text/javascript" src="../js/echarts/echarts\_pc2.js" ></script>

<!-- 引入 echarts.js -->

</head>

<body>

<!-- 为ECharts准备一个具备大小（宽高）的Dom -->

<div id="main" style="width: 600px;height:400px;"></div>

<script type="text/javascript">

// 基于准备好的dom，初始化echarts实例

**var** myChart = echarts.init(document.getElementById('main'));

// 使用刚指定的配置项和数据显示图表。

myChart.setOption(option);

</script>

</body>

</html>

### 4.1.3 CSS代码

#my\_div{

*width*: 1000px;

*height*: 1000px;

*border*: 1px solid yellow;

*margin*: auto;

}

#foot{

*margin*: auto;

*width*: 1000px;

*border*: 1px solid black;

*text-align*: center;

}

### 4.1.4 js代码

setTimeout(*function* () {

*option* = {

legend: {},

tooltip: {

trigger: 'axis',

showContent: false

},

dataset: {

source: [

['product', '2012', '2013', '2014', '2015', '2016', '2017'],

['java', 41.1, 30.4, 65.1, 53.3, 83.8, 98.7],

['python', 86.5, 92.1, 85.7, 83.1, 73.4, 55.1],

['ruby', 24.1, 67.2, 79.5, 86.4, 65.2, 82.5],

['c++', 55.2, 67.1, 69.2, 72.4, 53.9, 39.1]

]

},

xAxis: {type: 'category'},

yAxis: {gridIndex: 0},

grid: {top: '55%'},

series: [

{type: 'line', smooth: true, seriesLayoutBy: 'row'},

{type: 'line', smooth: true, seriesLayoutBy: 'row'},

{type: 'line', smooth: true, seriesLayoutBy: 'row'},

{type: 'line', smooth: true, seriesLayoutBy: 'row'},

{

type: 'pie',

id: 'pie',

radius: '30%',

center: ['50%', '25%'],

label: {

formatter: '{b}: {@2012} ({d}%)'

},

encode: {

itemName: 'product',

value: '2012',

tooltip: '2012'

}

}

]

};

myChart.on('updateAxisPointer', *function* (*event*) {

*var* xAxisInfo = *event*.axesInfo[0];

if (xAxisInfo) {

*var* dimension = xAxisInfo.value + 1;

myChart.setOption({

series: {

id: 'pie',

label: {

formatter: '{b}: {@[' + dimension + ']} ({d}%)'

},

encode: {

value: dimension,

tooltip: dimension

}

}

});

}

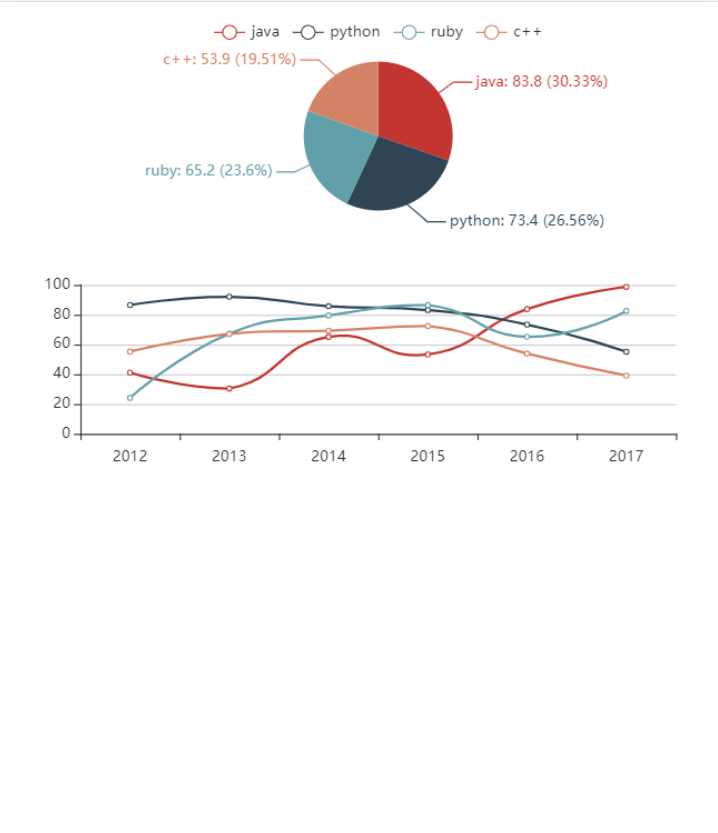
});

myChart.setOption(option);

});

## 4.2 APP端页面效果

### 4.2.1 页面效果截图



### 4.2.2 HTML代码

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />

<title></title>

<!--<script type="text/javascript">

document.addEventListener('plusready', function(){

//console.log("所有plus api都应该在此事件发生后调用，否则会出现plus is undefined。"

});

</script>-->

<meta charset="utf-8">

<title>ECharts</title>

<script src="js/mui.js"type="text/javascript" charset="UTF-8"></script>

<script src="js/mui.min.js"type="text/javascript" charset="UTF-8"></script>

<script type="text/javascript" src="js/echarts.js" ></script>

<script type="text/javascript" src="js/echarts/echarts\_pc2.js" ></script>

<!-- 引入 echarts.js -->

</head>

<body>

<!-- 为ECharts准备一个具备大小（宽高）的Dom -->

<div id="main" style="*width*: 600px;*height*:400px;"></div>

<script type="text/javascript">

// 基于准备好的dom，初始化echarts实例

*var* myChart = echarts.init(*document*.getElementById('main'));

// 使用刚指定的配置项和数据显示图表。

myChart.setOption(*option*);

</script>

</body>

</html>

### 4.2.3 CSS代码

#my\_div{

*width*: 1000px;

*height*: 1000px;

*border*: 1px solid yellow;

*margin*: auto;

}

#foot{

*margin*: auto;

*width*: 1000px;

*border*: 1px solid black;

*text-align*: center;

}

### 4.2.4 js代码

setTimeout(*function* () {

*option* = {

legend: {},

tooltip: {

trigger: 'axis',

showContent: false

},

dataset: {

source: [

['product', '2012', '2013', '2014', '2015', '2016', '2017'],

['java', 41.1, 30.4, 65.1, 53.3, 83.8, 98.7],

['python', 86.5, 92.1, 85.7, 83.1, 73.4, 55.1],

['ruby', 24.1, 67.2, 79.5, 86.4, 65.2, 82.5],

['c++', 55.2, 67.1, 69.2, 72.4, 53.9, 39.1]

]

},

xAxis: {type: 'category'},

yAxis: {gridIndex: 0},

grid: {top: '55%'},

series: [

{type: 'line', smooth: true, seriesLayoutBy: 'row'},

{type: 'line', smooth: true, seriesLayoutBy: 'row'},

{type: 'line', smooth: true, seriesLayoutBy: 'row'},

{type: 'line', smooth: true, seriesLayoutBy: 'row'},

{

type: 'pie',

id: 'pie',

radius: '30%',

center: ['50%', '25%'],

label: {

formatter: '{b}: {@2012} ({d}%)'

},

encode: {

itemName: 'product',

value: '2012',

tooltip: '2012'

}

}

]

};

myChart.on('updateAxisPointer', *function* (*event*) {

*var* xAxisInfo = *event*.axesInfo[0];

if (xAxisInfo) {

*var* dimension = xAxisInfo.value + 1;

myChart.setOption({

series: {

id: 'pie',

label: {

formatter: '{b}: {@[' + dimension + ']} ({d}%)'

},

encode: {

value: dimension,

tooltip: dimension

}

}

});

}

});

myChart.setOption(*option*);

});

## 4.2查询的 dao函数

package com.headphone\_net.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import com.head\_phone.beans.Code;

import com.headphone\_src.utils.DBUtil;

public class CodedaoImpl {

public ArrayList<Code> searchCode()

{

ArrayList<Code> codelist=new ArrayList<Code>();

DBUtil dbUtil=new DBUtil();

Connection conn= (Connection) dbUtil.getConn();

ResultSet rs=null;

try {

String sql="select \* from t\_code where code\_name=?";

PreparedStatement pstmt=conn.prepareStatement(sql);

pstmt=conn.prepareStatement(sql);

rs=pstmt.executeQuery();

while (rs.next()) {

int code\_id=rs.getInt("code\_id");

String code\_name=rs.getString("code\_name");

int code\_use=rs.getInt("code\_use");

Code code=new Code();

code.setCode\_id(code\_id);

code.setCode\_name(code\_name);

code.setCode\_use(code\_use);

codelist.add(code);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

try {

rs.close();

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return codelist;

}

}

## 4.3 查询的servlet函数

package com.headphone\_net.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.ArrayList;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.head\_phone.beans.Code;

import com.headphone\_net.dao.CodedaoImpl;

import com.google.gson.Gson;

public class CodeServlet extends HttpServlet{

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

super.doPost(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

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

if(para.equals("search\_code"))

{

searchCode(request,response);

System.out.println("6666");

}

}

private void searchCode(HttpServletRequest request, HttpServletResponse response) {

// TODO Auto-generated method stub

ArrayList<Code> codelist=(new CodedaoImpl()).searchCode();

Gson gson=new Gson();

String json\_arr=gson.toJson(codelist);

try {

PrintWriter out=response.getWriter();

out.println(json\_arr);

out.flush();

out.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

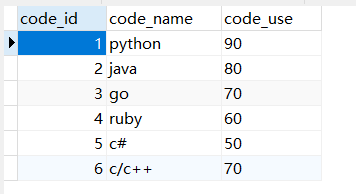
}

}

}

## 4.4 数据库截图





# 五、思考题

1、什么是JSON?

2、什么是MUI?

3、Echart哪个公司开发的？主要功能包含什么？