# 5 PHP访问数据库的实现

## 5.1 PHP与数据库的连接实现

由于不同结构的网络数据库对数据库的访问方式和操作效率区别不同。对于WEB数据库技术主要就是怎样通过网络访问服务器端数据库。目前流行就是用执行HTML、Script、和SQl命令，通过ODBC来访问数据库。客户端访问数据库步骤如下：

HTTP请求 SQL请求

服务器

数据库服务器

Web浏览器

HTML页数据结构

图5-1数据流图

第一步：用户在页面上点击提交按钮，把表单内容传送到服务器；

第二部：服务器通过连接到数据库，执行SQL命令；

第三步：SQL执行结果以HTML形式传送给用户浏览器。

由于php语言可以轻松访问WEB数据库。其成绩管理系统采用模块设计，则数据库连接包含文件由实现，如下：

public function open($savePath, $sessName)

{

$this->lifeTime= C('SESSION\_EXPIRE') ? C('SESSION\_EXPIRE') : ini\_get('session.gc\_maxlifetime');

$this->sessionTable=C('SESSION\_TABLE')?C('SESSION\_TABLE') : C("DB\_PREFIX") . "session";

//分布式数据库

$host = explode(',', C('DB\_HOST'));

$port = explode(',', C('DB\_PORT'));

$name = explode(',', C('DB\_NAME'));

$user = explode(',', C('DB\_USER'));

$pwd = explode(',', C('DB\_PWD'));

if (1 == C('DB\_DEPLOY\_TYPE')) {

//读写分离

if (C('DB\_RW\_SEPARATE')) {

$w = floor(mt\_rand(0, C('DB\_MASTER\_NUM') - 1));

if (is\_numeric(C('DB\_SLAVE\_NO'))) {//指定服务器读

$r = C('DB\_SLAVE\_NO');

} else {

$r=floor(mt\_rand(C('DB\_MASTER\_NUM'), count($host) - 1));

}

//主数据库链接

$dbSel = mysqli\_select\_db(

$hander,

isset($name[$w]) ? $name[$w] : $name[0]

);

if (!$hander || !$dbSel)

return false;

$this->hander[0] = $hander;

//从数据库链接

$hander = mysqli\_connect(

$host[$r] . (isset($port[$r]) ? ':' . $port[$r] : ':' . $port[0]),

isset($user[$r]) ? $user[$r] : $user[0],

isset($pwd[$r]) ? $pwd[$r] : $pwd[0]);

$dbSel = mysqli\_select\_db(

$hander,

isset($name[$r]) ? $name[$r] : $name[0]

);

if (!$hander || !$dbSel)

return false;

$this->hander[1] = $hander;

return true;

}

}

//从数据库链接

$r = floor(mt\_rand(0, count($host) - 1));

$hander = mysqli\_connect(

$host[$r] . (isset($port[$r]) ? ':' . $port[$r] : ':' . $port[0]),

isset($user[$r]) ? $user[$r] : $user[0],

isset($pwd[$r]) ? $pwd[$r] : $pwd[0]

);

$dbSel = mysqli\_select\_db(

$hander,

isset($name[$r]) ? $name[$r] : $name[0]

);

if (!$hander || !$dbSel)

return false;

$this->hander = $hander;

return true;

}

/\*\*

\* 关闭Session

\* @access public

\*/

public function close()

{

if (is\_array($this->hander)) {

$this->gc($this->lifeTime);

return (mysqli\_close($this->hander[0]) && mysqli\_close($this->hander[1]));

}

$this->gc($this->lifeTime);

return mysqli\_close($this->hander);

此代码为数据库的连接与退出，连接的xuesheng.sql

## 5.2 数据库增加记录实现

对于学生的成绩与信息，都需要教师登录才可以进入相应界面进行增加。最后都将存入数据库。接下来是成绩管理系统模块展示。

首先是教师进入教师界面进行学生成绩的添加。区分学生与教师就由教师与学生的学号不同进而进入不同界面。下面是相应截图与代码。



图5-2登录图

上图界面就是登录界面，教师端由教师账号与密码登录，学生由学号与密码登录服务器通过post（Post是向服务器提交数据的一种请求指定的资源提交要被处理的数据POST比GET更安全，因为参数不会被保存在浏览器历史或web服务器日志中）方式进入服务器与数据库中数据相匹配，若一致则进入教师界面或学生的界面进行相应的操作。还有重置按钮可以渐变的方便教师与学生登录时的麻烦。

例如：（教师的账号与密码都为001）进入示范：



图5-3登录示例图

点击登录后就可以进入教师的管理界面。若教师登录账号与密码与数据库不一致时则会直接回到初始界面，则说明教师登录的账号与密码不匹配。重新输入，方可进入。

相应的代码：

<div class="login\_form">

<form action="{:U('Index:login')}" id="login\_form" method="post">

<div class="form-group">

<label for="j\_username" class="t">账&nbsp;&nbsp;&nbsp;&nbsp;号：</label>

<input id="email" value="" name="username" type="text" class="form-control x319 in"

autocomplete="off">

</div>

<div class="form-group">

<label for="j\_password" class="t">密　码：</label>

<input id="password" value="" name="password" type="password"

class="password form-control x319 in">

</div>

教师在登录界面输入相应的账号与密码通过session进行传递通过Index:login.php传递数据库。数据库根据request命令提交给数据库进行比对在返回命令。

进入教师界面后增加学生信息则相应点击增加成绩栏进入。形成下面的界面，在相应得界面输入即可。



图5-4教师增加学生信息的界面

进入后在输入相应学号成绩就自动加入一个学生信息与成绩。相应密码就是学号，学生可以在学生端界面进行相应修改。教师在增加信息这个界面通过增加学号、姓名，再依次输入相应学生高数、英语、数据结构、数据库的成绩！

例如：加入学生学号为052姓名为王大帅的学生成绩！



图5-5成绩增加示例图



图5-6成绩显示图

这样就可以向数据库中增加一个数据！

其代码如下：cd5f64965fdddcee253d83c997e866b8.php文件

<div class="banner">

<div class="bgh">

<div class="page">

<div id="logo">

<a href="../Index.aspx.html">

<img src="/li/Public/Images/Student/logo.gif" alt="" width="165" height="48" />

</a>

<div class="page">

<div class="box mtop">

<div class="leftbox">

<div class="l\_nav2">

<div class="ta1">

<strong>个人中心</strong>

<div class="leftbgbt">

</div>

</div>

<div class="cdlist">

<div><a href="<?php echo U('Index:stu\_info\_all');?>">学生信息</a></div>

<div><a href="#">学生反馈</a></div>

<div><a href="<?php echo U('Index:add\_stu');?>">增加信息</a></div>

<div><a href="<?php echo U('Index:delete\_stu');?>">删除信息</a></div>

<div><a href="<?php echo U('Index:edit\_stu');?>">修改信息</a></div>

</div>

<div class="ta1">

<strong>教务中心</strong>

<div class="leftbgbt2">

<div class="cdlist">

<div>

<a href="<?php echo U('Index:changepass');?>">密码修改</a></div>

<div>

<a href="<?php echo U('index:logout');?>">安全退出</a></div>

<div class="rightbox">

<h2 class="mbx">我的信息&gt; 我的成绩&nbsp;&nbsp;&nbsp;</h2>

<div class="cztable">

<form action="<?php echo U('Index:add\_stu\_fu');?>" method="post">

<label for="num">学&nbsp;&nbsp;&nbsp;号&nbsp;&nbsp;&nbsp;&nbsp;：</label>

<input type="text" id="num" name="stu\_num">

<label for="name">姓&nbsp;&nbsp;&nbsp;名&nbsp;&nbsp;&nbsp;&nbsp;

</label>

<input type="text" id="name" name="name">

<label for="math">高等数学：</label>

<input type="text" id="math" name="gaoshu">

<label for="engthis">英&nbsp;&nbsp;&nbsp;语：</label>

<input type="text" id="engthis"name="english">

<label for="dataStructure">数据结构：</label>

<input type="text" id="dataStructure"name="shujujiegou">

<label for="dataBase">数&nbsp;据&nbsp;库&nbsp;&nbsp;：</label>

<inputtype="text"id="dataBase"name="shujuku">

<br>

<br>

<button type="submit">提交</button>

着重说一下，其代码没有考虑加入学生密码只有相应学号的输入与成绩的输入。故我重定向为密码与学号一致。在这里我没设置相应的限制，只要是数字就可以进入。关于学生密码学生可以进入自己界面进行修改。还有一部分是学生信息的增加代码。在php的内置对象session中赋值给相应的变量。这里值得说的是，用读取的方法把session的数据读出，跳转在stu\_info\_all.php对上述进行传递进数据库。一般采用将表单中的数据提交，提交进数据库就可以实现相应操作。

## 5.3 数据库修改记录设计的实现

由于教师疏忽或是学生表现优异，成绩需要实现可以时时修改的属性，所以我在教师端加入了学生成绩修改功能：

首先是学生修改功能界面：



图5-7修改图

教师点击修改信息界面，进入后登录教师号，再次输入学生现在的成绩即可，在上述界面填写修改后修改的成绩后点击提交就可以通过submit属性把相关数据修改进数据库。

例如:



图5-8信息展示图

由此可以看出学生信息里有个学号为052名为王大帅的学生所有成绩都为0。所以我们要加入他的成绩（分别为100、105、90、85）：

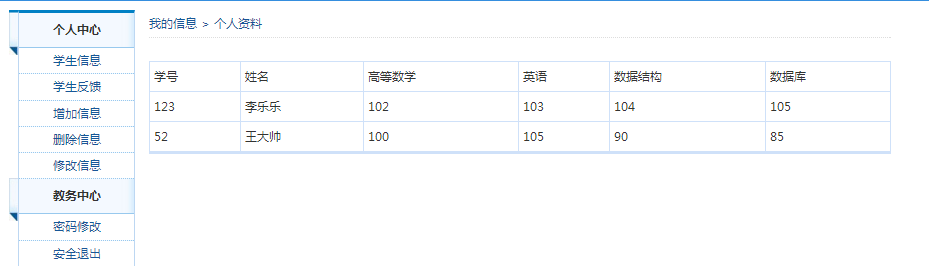


图5-9修改信息图

图5-10提交显示

点击提交就可以实现数据修改进入数据库。

当修改学生信息时，输入的账号与姓名不符合时即数据与数据库中的数据不匹配时则返回初试界面需重新进入修改界面再进行修改。只要当账号与密码与数据库中的数据一致时才可以对数据库中的数据进行修改。上述这就是一个完整的教师修改学生信息的方式。

其代码如下：

<script type="text/javascript">

function submitMail() {

var mtitle = "修改";

</textarea></div></div>"<h2 class="mbx">我的信息&gt; 我的成绩&nbsp;&nbsp;&nbsp;</h2><div class="cztable">

<form action="{:U('Index:edit\_stu\_fu')}" method="post">

<label for="num">学&nbsp;&nbsp;&nbsp;号</label>

<label for="name">姓&nbsp;&nbsp;&nbsp;名&nbsp;&nbsp;&nbsp;&nbsp;：</label>

<input type="text" id="name" name="name">

<label for="math">高等数学：</label>

<input type="text" id="math" name="gaoshu">

<label for="engthis">英&nbsp;&nbsp;&nbsp;语

<input type="text" id="engthis"name="english">

<label for="dataStructure">数据结构：</label>

<input type="texid="dataStructure"name="shujujiegou">

<label for="dataBase">数&nbsp;据&nbsp;库&nbsp;&nbsp;：</label>

<input type="text" id="dataBase"name="shujuku">

<button type="submit">提交</button>

StudentCompain.insertCompain('', mtitle, 5, f.objeCont, function (data) { var obj = $.parseJSON(data);

var resultObj = false;

if (obj.ok) {

$.jBox.tip("成功提交修改！"); $.jBox(html, { title: "修改的", submit: submit });

}

</script

以上代码就是教师修改信息的方式。通过request对象读取表单上的修改的数据，用SQl语句将指针指向当前修改的的用户， edit\_stu\_fu.php更新数据库中的记录的数据。通过点击修改界面的修改按钮，使用update命令就可以实现对数据库中的数据进行修改，这就是相关代码的含义。

## 5.4 数据库删除记录设计的实现

毕业后的学生的成绩也该从数据库删去。如果不删除，学生信息太多，耽误教师的管理增加冗余度。所以增加了删除功能：

下面是删除功能界面:



图5-11删除显示图

当新学生进入我校后，如果不删除他们的成绩在学生成绩管理数据库中就会形成数据冗余的情况发生，这样教师无法正常工作，学生也无法正常查到自己真实的成绩，形成错误的信息。故我加入了把数据库中的数据删除的功能。上述界面中教师只需点击删除信息按钮，进入删除界面。在学号输入框输入要删除的学生相应学号点击删除按钮就可以把数据库中的相应学生信息进行删除。例如:

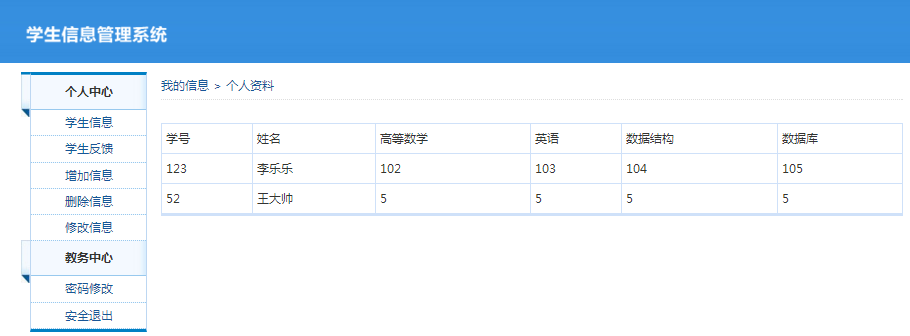


图5-12信息显示图

由此可以看出有个学号为052的学生相应成绩。当此学生离校后我们要删除该学生的相应成绩。



图5-13删除按钮显示图

在此输入学号进行点击删除按钮：



图5-14删除后信息显示图

现在数据库中只剩下学号为123的学生成绩。

由于这个删除只需要输入账号，所以安全性有点疏忽。但大大增加了教师的实现程度。这个只需输入要删除的学生账号即可。若输入的学生账号数据库中不存在时即为输错则返回初始界面，若还需删除在需输入正确账号就可以实现要删除的那个学生的成绩。这便是一个完整删除的过程。

相应删除代码如下：

<div class="rightbox">

<h2 class="mbx">我的信息&gt; 我的成绩;</h2>

<div class="cztable">

<form action="{:U('Index:delete\_stu\_fu')}" method="post">

<label for="num">学号：</label>

<input type="text" id="num" name="stu\_num">

<button type="submit">删除</button>

</form>

</div>

该代码实现教师登录后再写入新的学生成绩后，点击选项后传给后面的处理文件delete\_stu\_fu.php，如果是选择是多项都给的话，传送学号时，各个学号中间用逗号分隔；在delete\_stu\_fu.php中，在执行SQl中的删除语句使得数据库中的数据得以删除。

## 5.5 数据库查询功能实现

本成绩管理系统主要突出的就是其高效性，与高效率性。数据查询对于教师与学生是不同的，教师是可以看到所有的学生的信息。对于学生就是只看到自己的信息就可以了。这就用了一个type类型就行分隔，若是学生端只体现自己的成绩即可，当是教师时就可以看到全部的信息。本系统就是从登陆时就对教师与学生进行了区分，当教师登录后直接可以看每个学生的成绩信息。学生端进入后仅可以查看自己的成绩与信息。下面首先是教师端的查询实现展示：





图5-15 数据库显示图

此图展示了教师凭介自己的账号与密码登录教师的平台进行管理学生，这里没有专门的选项来区分学生与教师。只需账号与数据库中的数据相符一致时就可以进入教师界面。数据库中有个登录表为（user）中的type为区分是教师还是学生的根据只有当账号与密码统一时且type为00000001时方可进入教师端。进入后为下图：

教师界面



图5-16教师界面图

在此页面教师可以查看学生的相关信息。教师在这个界面首先可以查看全部学生成绩、可以查看学生给教师发的信息、可以查看学生的反馈信息、与增加学生信息、删除信息、修改信息、密码修改、安全退出的按钮。这里只说查询的学生信息按钮。如下图可以看到所有学生的学号、姓名、各科的学习成绩。

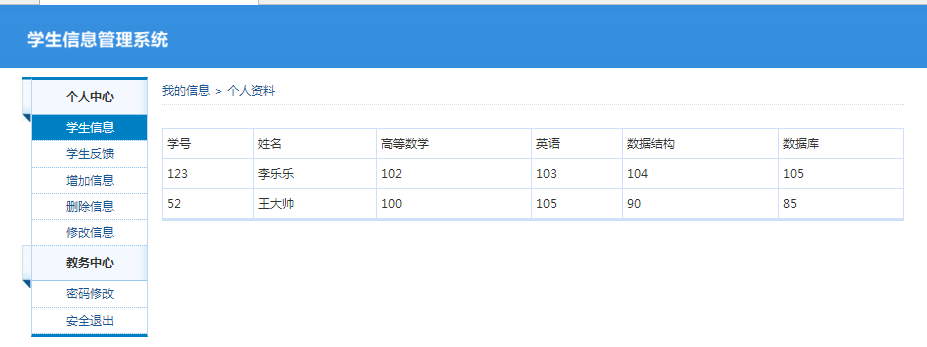


图5-17成绩显示图

下面给出相应的代码：

|  |  |
| --- | --- |
|  | <a href="[../Index.aspx.html](http://localhost/test/index.php/Home/Index.aspx.html" \t "_blank)"> |
|  | <h2 class="mbx">我的信息&gt; 个人资料&nbsp;&nbsp;&nbsp;</h2> |
|  | <div class="cztable"> |
|  | <table width="100%" cellpadding="0" cellspacing="0"> |
|  | <tr> |
|  | <td>学号</td> |
|  | <td>姓名</td> |
|  | <td>高等数学</td> |
|  | <td>英语</td> |
|  | <td>数据结构</td> |
|  | <td>数据库</td> |
|  | <td>123</td> |
|  | <td>李乐乐</td> |
|  | <td>102</td> |
|  | <td>103</td> |
|  | <td>104</td> |
|  | <td>105</td>< |
|  | <td>52</td> |
|  | <td>王大帅</td> |
|  | <td>100</td> |
|  | <td>105</td> |
|  | <td>90</td> |
|  | <td>85</td> |
|  |  |
|  | </table> |
|  |  |

相应代码就是根据Index.aspx.html通过连接数据库传到前端界面学生的成绩。首先是用户的包含语句，接着读取用户体检的查询关键，这个是内部数据库的具体实现。通过SQL语句实现用户怎么实现提取数据的，最终实现系统查询时可自行到全部学生的成绩展示。

相应的也有学生端的查询的功能：

学生端的登录：



图5-17学生登录图

当学生输入自己的学号与密码后经过数据库的验证，若与之不匹配则返回原始界面进行重新登录。若匹配则根据数据中的type类型进行判断是教师还是学生进行进入不同界面显示不同功能。当判断为学生且通过时进入下面这个属于学生的界面：



图5-18学生界面图

此图为学生端的初始界面显示了学生的功能界面。主要的还是可以查看我的成绩，我的信息、我的异议主要就是我的对成绩的一些看法一类的。



图5-20学生信息图

相应的源码：

<div class="rightbox">

<h2 class="mbx">我的信息&gt; 个人资料&nbsp;&nbsp;&nbsp;</h2>

<div class="cztable">

<table width="100%" cellpadding="0" cellspacing="0">

<?php if(is\_array($info\_list)): $i = 0; $\_\_LIST\_\_ = $info\_list;if( count($\_\_LIST\_\_)==0 ) : echo "" ;else: foreach($\_\_LIST\_\_ as $key=>$vo): $mod = ($i % 2 );++$i;?><tr>

<td align="right" width="80">姓名：</td>

<td><?php echo ($vo["name"]); ?>&nbsp;</td>

<td align="right" width="90">学号：</td>

<td><?php echo ($vo["stu\_num"]); ?>&nbsp;</td>

<td><?php echo ($vo["sex"]); ?>&nbsp;</td>

<td align="right">系别：</td>

<td><?php echo ($vo["college"]); ?>&nbsp;</td>

<td align="right">报考类别：</td>

<td><?php echo ($vo["bk\_type"]); ?>&nbsp;</td>

<td align="right">学校：</td>

<td><?php echo ($vo["school"]); ?>&nbsp;</td>

<td align="right">专业：</td>

<td><?php echo ($vo["major"]); ?>&nbsp;</td>

<td align="right">原专业：</td>

<td><?php echo ($vo["his\_major"]); ?>&nbsp;</td>

<td align="right">政治面貌：</td>

<td><?php echo ($vo["zzmm"]); ?>&nbsp;</td>

<td align="right">注册批次：</td>

<td><?php echo ($vo["sign"]); ?>&nbsp;</td>

<td align="right">学制：</td>

<td><?php echo ($vo["study"]); ?>&nbsp;</td>

<td align="right">状态：</td>

<td><?php echo ($vo["status"]); ?>&nbsp;</td>

</tr><?php endforeach; endif; else: echo "" ;endif; ?>

这个就是个人信息的显示，所有注册过的信息。是通过用户的登录时的信息进行与数据库进行匹配再调用相关信息进行与数据库中的数据进行连接，通过SQL命令search语句实现查询自己的（姓名、学号、性别、系别、学校、专业、政治面貌等信息）。再次还可以查询自己的成绩：

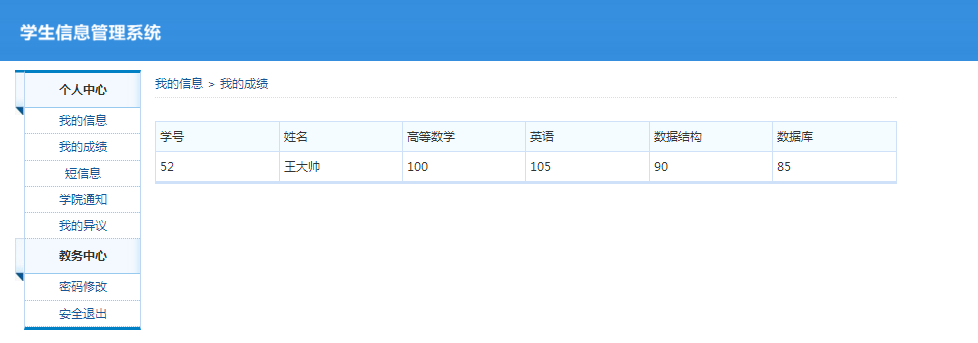


图5-21学生成绩图

如果以学生身份登录，通过登录时就已经确定是学生。通过SQL命令语句进行与数据库中的当前的学生进行数据库与前端界面进行连接，实现当前该学生与数据库中的该学生进行链接实现查询自己成绩的功能。显示中仅体现了该学生的学号，姓名、以及各科成绩的展示。不会再显示其余学生这是由相应命令决定保证了学生的唯一性。

相关代码:

<div><a href="/test/index.php/Home/Index/my\_info.html">我的信息</a></div>

<div><a href="/test/index.php/Home/Index/my\_grade.html">我的成绩</a></div><h2 class="mbx">我的信息&gt; 我的成绩&nbsp;&nbsp;&nbsp;</h2>

<div class="cztable">

<table width="100%" cellpadding="0" cellspacing="0">

<tr>

<td>学号</td>

<td>姓名</td>

<td>高等数学</td>

<td>英语</td>

<td>数据结构</td>

<td>数据库</td>

<td>52</td>

<td>王大帅</td>

<td>100</td>

<td>105</td>

<td>90</td>

</table>

</div>

此代码主要就是说明了上图中的个人成绩的体现是如何体现的。通过界面的学号与数据库匹配，通过SQL语句与个人进行数据连接。最后通过chengji.php进行数据的传递实现学生成绩的功能！

不论教师还是学生的相关信息的查询主要还是在登录界面要与数据库中的数据相一致进入才可以显示相应身份的成绩，所以一定得保障登录时的正确性才可以做到数据的安全性。

# 6 数据库中的密码修改实现

密码对于我们每个人来说都只关重要，如果密码丢失将发生不可估量的错误。若密码简单容易被人攻破也会发生相应的错误。例如教师的密码丢失，学生万一知道密码后改变相应的成绩就会发生不可估量的错误。所以在我的成绩管理系统中加入了密码修改相关的功能。

## 6.1 教师端的密码修改

下面是教师界面的密码修改：



图6-1 教师界面图

上图为教师进入教师界面的管理界面，第三栏中的密码选项。点击密码修改按钮进入下面界面。



图6-2密码修改图

修改界面在上图中显示的旧密码、新密码、确认密码。有着旧密码选项必须要求修改密码的人得知道密码才可以修改。这就确保了账号的安全性不至于被外人所利用进行坏事。若输入旧密码后不输入新密码是无法点击确认密码。若不输入输入相关密码时点击时则会使密码进入死密码无法破解，只有管理员进入后台数据库中才可以编辑修改密码。

代码如下：

<a href="../../Index.aspx.html">

<div class="rightbox">

<h2 class="mbx">我的信息&gt; 密码修改</h2>

<div class="cztable">

<form action="{:U('Index:changeps')}" method="post">

<table border="0" cellspacing="0" cellpadding="0" width="500px" style="margin:30px auto 0px auto;">

<tr align="center">

<th style="width:20%; text-align:left;">旧密码：</th>

<td style="width:70%; text-align:left;"><input id="txtOldPwd" value="" type="password" class="input\_2 txtinput1" /></td>

<tr align="center">

<th style="width:20%; text-align:left;">新密码：</th>

<td style="width:70%; text-align:left;"><input id="txtNewPwd" value="" type="password" class="input\_2 txtinput1" name="password" />&nbsp;&nbsp;6~16个字符，区分大小写</td>

<tr align="center">

<th style="width:20%; text-align:left;">确认新密码：</th>

<td style="width:70%; text-align:left;"><input id="txtConfirmNewPwd" value="" type="password" class="input\_2 txtinput1" /></td>

<td colspan="2" style="text-align:center;"><input type="submit" id="btnSubmit" value="确认修改" class="input2" /></td>

其代码的含义就是上述输入后，通过password.php连接到数据库，通过SQL命令语句由.php中的语句将相应数据通过SQl命令将数据传入到后台数据库中将原来数据空中的数据进行覆盖即更新操作。

## 6.2 学生端的密码修改

其次也有学生端的密码修改，因为学生端的初始创建时学号与密码一起都是初始的，所以很容易被人利用。所以在学生端我也加入了修改密码的功能。下面就是学生端的密码修改：



图6-3 学生界面图

这个界面便是学生登陆后的初始界面，在此界面可以看出有密码修改的功能按钮，当登陆成功后，本系统会提醒你首先修改自己的密码的问题。若不进行密码修改问题，将不允许所有类型的操作。所以该成绩管理系统保障了学生的信息安全问题。所以只有修改密码情况下才可以进行下一步的的操作。

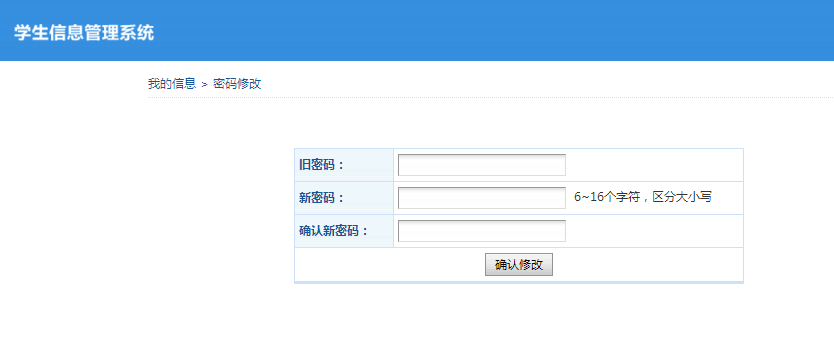
下图为学生进入密码修改后的界面：

图6-3学生密码修改

这里的规则与教师端的相同。修改界面在上图中显示的旧密码、新密码、确认密码。有着旧密码选项必须要求修改密码的人得知道密码才可以修改。这就确保了账号的安全性不至于被外人所利用进行坏事。若输入旧密码后不输入新密码是无法点击确认密码。若不输入输入相关密码时点击时则会使密码进入死密码无法破解，只有管理员进入后台数据库中才可以编辑修改密码，进行修改否则将不能进行登录使用！

代码与上同，只是传参数的位置不同。

这就是该成绩管理系统修改密码。