

《Web开发实践》

姓名：田晋宇 学号：2212039 班级：物联网工程

实验名称

Web开发实践

实验要求

复现课本第十章的实验三(10.3.5节):利用php, 编写简单的数据库插入、查询和删除操作的示例。基于课本的完整例子, 进一步了解WEB开发的细节。

实验过程

1. 安装PHPnow

在安装过程中发现, 在win7,win10版本存在管理员机制, 必须使用管理员权限运行Init.cmd, 否则会运行失败, 可以使用Windows XP系统进行本次实验, 选择PHPnow的Apache和Mysql版本:



```
选择 Apache 版本
20 - Apache 2.0.63 <推荐>
22 - Apache 2.2.16
-> 请选择: 20

选择 MySQL 版本
50 - MySQL 5.0.90 <推荐>
51 - MySQL 5.1.50
-> 请选择: 50
```

打开默认界面 (<http://127.0.0.1/index.php>)

127.0.0.1

Let's PHP now !

为何只能本地访问?

此服务器互联网 IP

111.33.78.4

Server Information	
SERVER_NAME	127.0.0.1
SERVER_ADDR:PORT	127.0.0.1:80
SERVER_SOFTWARE	Apache/2.0.63 (Win32) PHP/5.2.14
PHP_SAPI	apache2handler
php.ini	C:\PHPnow-1.5.6\php-5.2.14-Win32\php-apache2handler.ini
网站主目录	C:\PHPnow-1.5.6\htdocs
Server Date / Time	2024-05-27 12:52:13 (+08:00)
Other Links	phpinfo() phpMyAdmin

PHP 组件支持	
Zend Optimizer	Yes / 3.3.3
MySQL 支持	Yes / client lib version 5.0.90
GD library	Yes / bundled (2.0.34 compatible)
eAccelerator	No

MySQL 连接测试			
MySQL 服务器	localhost	MySQL 数据库名	test
MySQL 用户名	root	MySQL 用户密码	
			连接

Valid XHTML 1.0 Strict / Copyleft ! 2007-? by PHPnow.org

输入密码检查数据库连接是否正常：

MySQL 连接测试			
MySQL 服务器	localhost	MySQL 数据库名	test
MySQL 用户名	root	MySQL 用户密码	
			连接

MySQL 测试结果	
服务器 localhost	OK (5.0.90-community-nt)
数据库 test	OK

2. 利用PHP，编写简单的数据库插入、查询和删除操作的示例复现

新建数据库TestDB



在数据库中创建数据表News (newsid, topic, content)

localhost ▶ testDB ▶ news			
字段	newsid	topic	content
类型	INT	VARCHAR	TEXT
长度/值 ¹		50	
默认 ²	无	无	无
整理			
属性			
空	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
索引	PRIMARY	---	---
AUTO_INCREMENT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
注释			

userinfo (username, password)

localhost ▶ testDB ▶ userinfo

字段	username	password
类型②	VARCHAR	VARCHAR
长度/值 ¹	30	30
默认 ²	无	无
整理		
属性		
空	<input type="checkbox"/>	<input type="checkbox"/>
索引	PRIMARY	---
AUTO_INCREMENT	<input type="checkbox"/>	<input type="checkbox"/>
注释		













均创建成功







localhost ▶ testdb ▶ userinfo

浏览 结构 SQL 搜索 插入 导出 导入 操作 清空 删除

✓ 创建数据表 'testdb`.`userinfo` 成功。

```
CREATE TABLE `testdb`.`userinfo` (
  `username` VARCHAR( 30 ) NOT NULL ,
  `password` VARCHAR( 30 ) NOT NULL ,
  PRIMARY KEY ( `username` )
) ENGINE = MYISAM ;
```

字段	类型	整理	属性	空	默认	额外	操作
<input type="checkbox"/> username	varchar(30)	latin1_swedish_ci		否	无		     
<input type="checkbox"/> password	varchar(30)	latin1_swedish_ci		否	无		     

↑ 全选 / 全不选 选中项:      

打印预览 规划表结构

添加 1 个字段 于表结尾 于表开头 于 之后 username 执行

向表中插入管理员数据admin:

localhost ▶ testDB ▶ userinfo

结构 SQL 搜索 查询 导出 导入 操作 权限 删除

✓ 已插入 1 行。

```
INSERT INTO `testDB`.`userinfo` (
  `username` ,
  `password`
)
VALUES (
  'admin' , 'admin'
);
```

新建html文档，准备编写html代码，新建html文件，准备编写html和PHP代码。



3. html 以及 php 代码编写:

下面进行html 以及 php 代码编写:

■ login.html

```
<html>

<body>

<form id="form1" name="form1" method="post" action="loginok.php">

<table width="900" border="0" cellspacing="0" cellpadding="0">

<tr> <td height="20">username</td>

<td height="20"><label>

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

</label></td>

</tr>

<tr>

<td height="20">password</td>

<td height="20"><label>

<input name="pwd" type="password" id="pwd" />

</label></td>

</tr>
```

```
<tr>

<td height="20"> </td>

<td height="20"><label>

<input type="submit" name="Submit" value="提交" />

</label></td>

</tr>

</table>

</form>

</body>

</html>
```

■ loginok.php

```
<?php

$loginok=0;

$conn=mysql_connect("localhost", "root", "123456");

$username = $_POST['username'];

$pwd = $_POST['password'];

$SQLStr = "SELECT * FROM userinfo where username='$username' and
pwd='$pwd'";

echo $SQLStr;

$result=mysql_db_query("testDB", $SQLStr, $conn);

if ($row=mysql_fetch_array($result))//通过循环读取数据内容

{

$loginok=1;

}

// 释放资源

mysql_free_result($result);

// 关闭连接

mysql_close($conn);
```

```

if ($loginok==1)

{

?>

<script>

alert("login succes");

window.location.href="sys.php";

</script>

<?php

}

else{?>

<script>

alert("login failed");

history.back();

</script>

<?php

}

?>

```

■ news.php

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>主页</title>
</head>
<body>
<div align="center">
<table width="900" border="0" cellspacing="0" cellpadding="0">
<tr>
<td height="40"><form id="form1" name="form1" method="post"
action="loginok.php">
<div align="right">用户名:
<input name="username" type="text" id="username" size="12" />
密码:
<input name="password" type="password" id="password" size="12" />
<input type="submit" name="Submit" value="提交" />
</div>
</form>
</td>

```

```

</tr>
<tr>
<td><hr /></td>
</tr>
<tr>
<td height="300" align="center" valign="top"><p> </p>
<?php
$conn=mysql_connect("localhost", "root", "123456");
$newsid = $_GET['newsid'];$SQLStr = "select * from news where
newsid=$newsid";
$result=mysql_db_query("testDB", $SQLStr, $conn);
if ($row=mysql_fetch_array($result))//通过循环读取数据内容
{
// 定位到第一条记录
mysql_data_seek($result, 0);
// 循环取出记录
while ($row=mysql_fetch_row($result))
{
echo "$row[1]<br>";
echo "$row[2]<br>";
}
}
// 释放资源
mysql_free_result($result);
// 关闭连接
mysql_close($conn);
?>
</td>
</tr>
</table>
</div>
</body>
</html>

```

通过上述代码实现了一个简单的用户登录和新闻显示系统。首先，通过 HTML 表单收集用户名和密码，并将数据提交给 PHP 脚本 "loginok.php" 进行验证。如果验证通过，用户会看到登录成功的提示并跳转到 "sys.php" 页面；如果验证失败，会提示登录失败并返回登录页面。此外，页面还包含一个 PHP 脚本，通过新闻 ID 从数据库中检索并显示相应的新闻内容。

username

password

输入username: admin, password: admin

```
SELECT * FROM userinfo where username=' admin' and password=' admin'
```

显示登录成功



4. 用 php 编写简单的数据库插入、查询和删除操作的示例

■ sys.php

```
<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />

<title>主页</title>

</head>

<?php

$conn=mysql_connect("localhost", "root", "123456");

?>

<body>

<div align="center">

<table width="900" border="0" cellspacing="0" cellpadding="0">

<tr>

<td height="40"><form id="form1" name="form1" method="post"
action="add.php">

<div align="right">新闻标题:

<input name="topic" type="text" id="topic" size="50" />

<BR>
```


新闻内容:

```
<textarea name="content" cols="60" rows="8" id="content"></textarea><BR>

<input type="submit" name="Submit" value="添加" />

</div>

</form>

</td>

</tr>

<tr>

<td><hr /></td>

</tr>

<tr>

<td height="300" align="center" valign="top"><table width="600"
border="0" cellpadding="0" cellspacing="0">

<tr>

<td width="100" height="30"><div align="center">新闻序号</div></td>

<td><div align="center">新闻标题</div></td>

<td><div align="center">删除</div></td>

</tr><?php

$SQLStr = "select * from news";

$result=mysql_db_query("testDB", $SQLStr, $conn);

if ($row=mysql_fetch_array($result))//通过循环读取数据内容
{

// 定位到第一条记录

mysql_data_seek($result, 0);

// 循环取出记录

while ($row=mysql_fetch_row($result))
{
```

```

?>

<tr>

<td height="30"><div align="center"> <?php echo $row[0] ?> </div></td>

<td width="400"> <div align="center"> <?php echo $row[1] ?>

</div></td>

<td><div align="center"><a href="del.php?newsid=<?php echo $row[0] ?> "
>

删 除 </a>

</div></td>

</tr>

<?php

}

}

?>

</table></td>

</tr>

</table>

</div>

</body>

</html>

<?php

// 释放资源

mysql_free_result($result);

// 关闭连接

mysql_close($conn);

?>

```

这段代码是一个简单的新闻管理网页，通过PHP和MySQL实现。网页的主要功能包括添加新闻、显示新闻列表以及删除新闻。实现了以下功能：

添加新闻：通过表单提交新闻标题和内容到 add.php 脚本进行处理。

显示新闻列表：从数据库中读取新闻记录并显示在网页上。

删除新闻: 提供删除新闻的链接, 通过 `del.php` 脚本处理删除操作。

- **add.php**

```
<?php
$conn=mysql_connect("localhost", "root", "123456");
mysql_select_db("testDB");
$topic = $_POST['topic'];
$content = $_POST['content'];
$SQLStr = "insert into news(topic, content) values('$topic',
'$content')";
echo $SQLStr;
$result=mysql_query($SQLStr);
// 关闭连接
mysql_close($conn);
if ($result)
{
?>
<script>
alert("insert succes");
window.location.href="sys.php";
</script>
<?php
}
else{
?>
<script>
alert("insert failed");
history.back();
</script>
<?php
}
?>
```

这段代码实现了从表单获取用户输入的新闻标题和内容，并将其插入到数据库中。插入操作完成后，根据结果显示相应的提示信息，成功时跳转到新闻管理页面，失败时返回到输入页面。PHP 脚本负责处理数据库连接、数据插入操作及结果反馈。

执行结果如下：

新闻标题:

新闻内容:

添加

新闻标题

删除

插入新闻标题为hello, 新闻内容为world, 插入成功

新闻标题:

新闻内容:

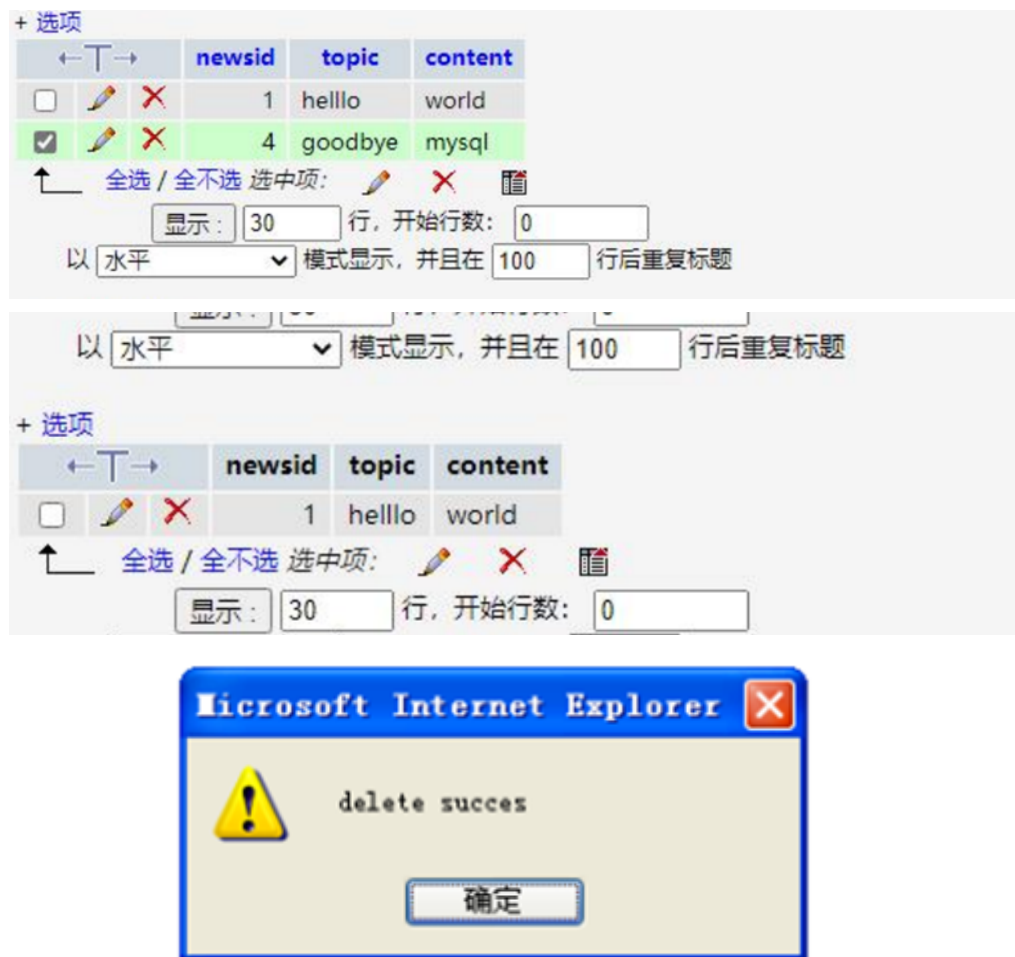


■ Del.php

```
<?php
$conn=mysql_connect("localhost", "root", "123456");
mysql_select_db("testDB");
$newsid = $_GET['newsid'];
$SQLStr = "delete from news where newsid=$newsid";echo $SQLStr;
$result=mysql_query($SQLStr);
// 关闭连接
mysql_close($conn);
if ($result)
{
?>
<script>
alert("delete succes");
window.location.href="sys.php";
</script>
<?php
}
else{
?>
<script>
alert("delete failed");
history.back();
</script>
<?php
}
?>
```

这段代码是一个用于删除新闻记录的PHP脚本。它通过接收 GET 请求中的 newsid 参数，从MySQL数据库中的 news 表中删除对应的新闻记录。

删除前后对比如下图所示：



心得体会

HTTP定义了与服务器交互的多种方法，其中最基本的有4种：GET、POST、PUT和DELETE。URL的全称是资源描述符，可以理解为一个URL地址描述了网络上的一个资源，而HTTP中的GET、POST、PUT、DELETE则对应了对这个资源的查询、更新、创建和删除操作。GET通常用于获取或查询资源信息，而POST则用于更新资源信息。由于早期系统不支持DELETE，因此PUT和DELETE使用较少。

具体来说，POST和GET的区别如下：

- GET请求的数据会附加在URL之后（即把数据放在HTTP协议头中），用?分隔URL和传输数据，参数之间用&连接，例如：login.action?name=sean&password=123。
- POST请求则将提交的数据放在HTTP包的包体中。POST的安全性比GET高。这里的安全性不仅指数据不容易被修改，还包括其他安全隐患。例如，通过GET提交数据，用户名和密码会明文出现在URL中，可能被浏览器缓存，其他人查看浏览器的历史记录就能获取到你的账号和密码。此外，使用GET提交数据还可能导致跨站请求伪造攻击（CSRF，Cross-site request forgery）。

这节课内容丰富，不仅涉及漏洞相关知识，还能同时学习多门语言。熟悉了PHP、HTML和MySQL这些语言，了解了网页编写的一些基础知识，使我对动态网页、静态网页、网页交互、POST和GET请求有了更深的理解。