

软件安全实验报告

姓名：辛浩然 学号：2112514 班级：信息安全、法学

1 实验名称

WEB 开发实践

2 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。基于课本的完整的例子，进一步了解 WEB 开发的细节。

3 实验过程

3.1 安装 PHPnow

PHPnow 下载解压后执行 Setup.cmd 初始化，即可得到一个 PHP + MySQL 环境。



```
管理员: C:\Windows\system32\cmd.exe
正在启动 MySQL 5.0 ...
Service successfully installed.
MySQL5_pn 服务正在启动:
MySQL5_pn 服务已经启动成功。
启动 MySQL 5.0 完成;

现在为 MySQL 的 root 用户设置密码. 重要! 请切记!
> 设置 root 用户密码: root

MySQL root 用户的新密码为 "root", 请切记!

全部完成!! 你将可以看到 PHPnow 的默认页面!
- 按任意键继续...
```

安装成功后启动 PHPnow。打开网页，访问 <http://127.0.0.1> 如下：

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:\Users\25123\Downloads\PHPnow-1.5.6\php-5.2.14-Win32\php-apache2handler.ini
网站主目录	C:/Users/25123/Downloads/PHPnow-1.5.6/htdocs
Server Date / Time	2023-05-09 08:29:31 (+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	
			<input type="button" value="连接"/>

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

3.2 创建数据库

进入 phpMyAdmin, 创建数据库 test, 创建表 1 news(newsid,topic,content) 和表 2 userinfo(username,password):



















localhost ▶ test ▶ news







浏览 结构 SQL 搜索 插入 导出 导入 操作 清空 删除

✓ 创建数据表 `test`.`news` 成功。

```
CREATE TABLE `test`.`news` (
  `newsid` INT NOT NULL,
  `topic` VARCHAR( 50 ) NOT NULL,
  `content` TEXT NOT NULL,
  PRIMARY KEY ( `newsid` )
) ENGINE = MYISAM ;
```

[编辑] [创建 PHP 代码]

	字段	类型	整理	属性	空	默认	额外	操作
<input type="checkbox"/>	newsid	int(11)			否	无		     
<input type="checkbox"/>	topic	varchar(50)	latin1_swedish_ci		否	无		     
<input type="checkbox"/>	content	text	latin1_swedish_ci		否	无		     

⬆ 全选 / 全不选 选中项:      













localhost ▶ test ▶ 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 `test`.`userinfo` 成功。

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

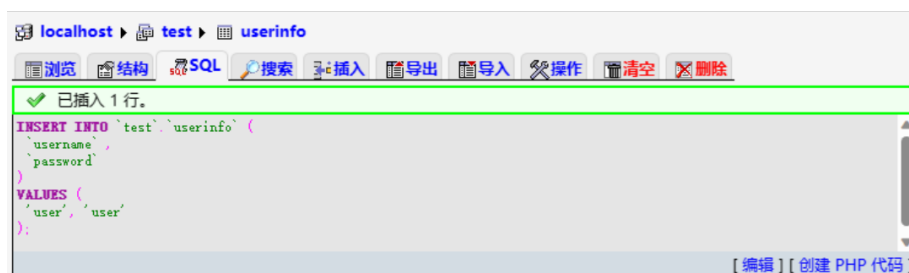
[编辑] [创建 PHP 代码]

	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:      

3.3 用户登录

首先，在数据库中插入用户信息：



编写 index.php，实现新闻显示和用户登录的功能。

```
1      <form id="form1" name="form1" method="post" action="loginok.php">  
2          <div align="right">用户名:  
3              <input name="username" type="text" id="username" size="12"  
4          />  
5              密 码:  
6              <input name="password" type="password" id="password" size=  
7          "12" />  
8              <input type="submit" name="Submit" value="提交" />  
9          </div>  
10     </form>
```

定义了一个表单，method="post"，指定了表单提交的 HTTP 方法为 POST，意味着表单数据将通过 HTTP 请求的主体传输。action="loginok.php" 指定了表单数据提交的目标 URL 为 loginok.php。当用户在用户名和密码输入字段中输入相应的信息，并点击提交按钮时，表单数据将被打包并发送到"loginok.php" 文件。

以下是 loginok.php 文件代码片段：

```
1      <?php  
2      $loginok = 0;  
3      $conn = mysql_connect("localhost", "root", "root");  
4      $username = $_POST['username'];  
5      $pwd = $_POST['password'];  
6      $SQLStr = "SELECT * FROM userinfo where username='$username' and  
7      password='$pwd'";  
8      echo $SQLStr;  
9      $result = mysql_db_query("test", $SQLStr, $conn);  
10     if ($row = mysql_fetch_array($result)) //通过循环读取数据内容  
11     {  
12         $loginok = 1;  
13     }  
14     // 释放资源  
15     mysql_free_result($result);  
16     // 关闭连接
```

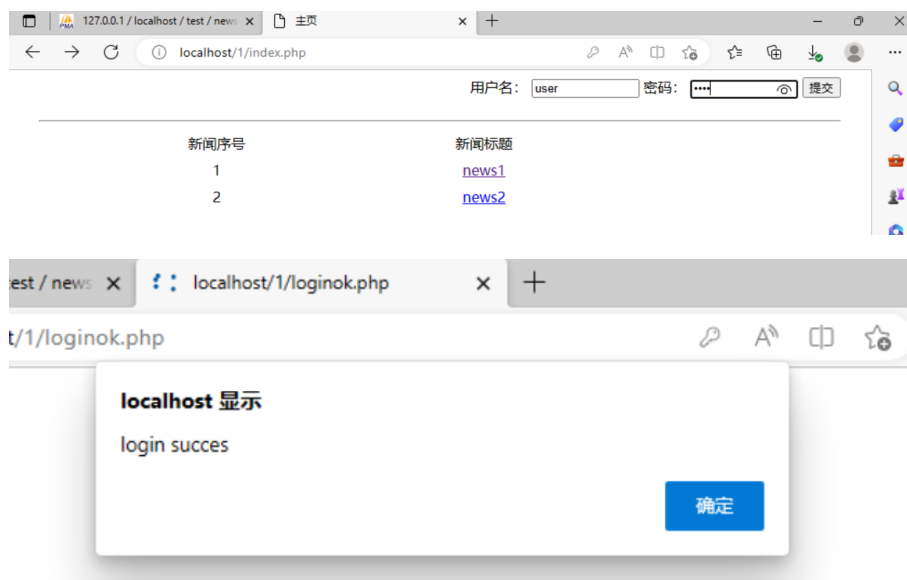
```

16     mysql_close($conn);
17     if ($loginok == 1) {
18         ?>
19         <script>
20             alert("login succes");
21             window.location.href = "sys.php";
22         </script>
23         <?php
24     } else {
25         ?>
26         <script>
27             alert("login failed");
28             history.back();
29         </script>
30         <?php
31     }
32
33     ?>
34

```

首先连接数据库，使用 POST 方式获取输入的用户名和密码。

构建 sql 语句查询数据库中是否有相同的用户名和密码，若是，则显示登陆成功，并跳转到 sys.php 界面执行下一步操作。否则，显示登陆失败，返回原界面。



3.4 新闻管理

在用户登录页面，展示新闻信息，展示新闻序号与新闻标题。点击新闻标题后会跳转到 news 界面查看相应的新闻内容。



编写 sys.php，实现新闻管理界面，管理数据库内新闻。

3.4.1 新闻插入

代码片段如下：

```

1  <form id="form1" name="form1" method="post" action="add.php">
2      <div align="right">新闻标题:
3          <input name="topic" type="text" id="topic" size="50" />
4          <BR>
5          新闻内容:
6          <textarea name="content" cols="60" rows="8" id="content"
></textarea><BR>
7          <input type="submit" name="Submit" value="添加" />
8      </div>
9  </form>
10

```

定义了一个表单，method="post"，指定了表单提交的 HTTP 方法为 POST，意味着表单数据将通过 HTTP 请求的主体传输。action="add.php" 指定了表单数据提交的目标 URL 为 add.php，即表单数据将发送到 add.php 文件进行处理。

<input> 标签用于接收用户输入，其中 name="topic" 指定了输入框的名称为 topic，type="text" 表示输入框为文本类型。

<textarea> 标签用于多行文本输入。

<input> 标签定义了一个提交按钮，其中 type="submit" 表示按钮为提交类型。当用户填写完表单中的新闻标题和内容后，点击提交按钮，表单数据将被封装到 HTTP 请求中，发送到 add.php 文件进行后续处理。

以下是 add.php 的代码：

```

1  <?php
2  $conn = mysql_connect("localhost", "root", "root");
3  mysql_select_db("test");
4  $topic = $_POST['topic'];
5  $content = $_POST['content'];
6  $SQLStr = "insert into news(topic, content) values('$topic', '
$content')";
7  echo $SQLStr;
8  $result = mysql_query($SQLStr);
9

```

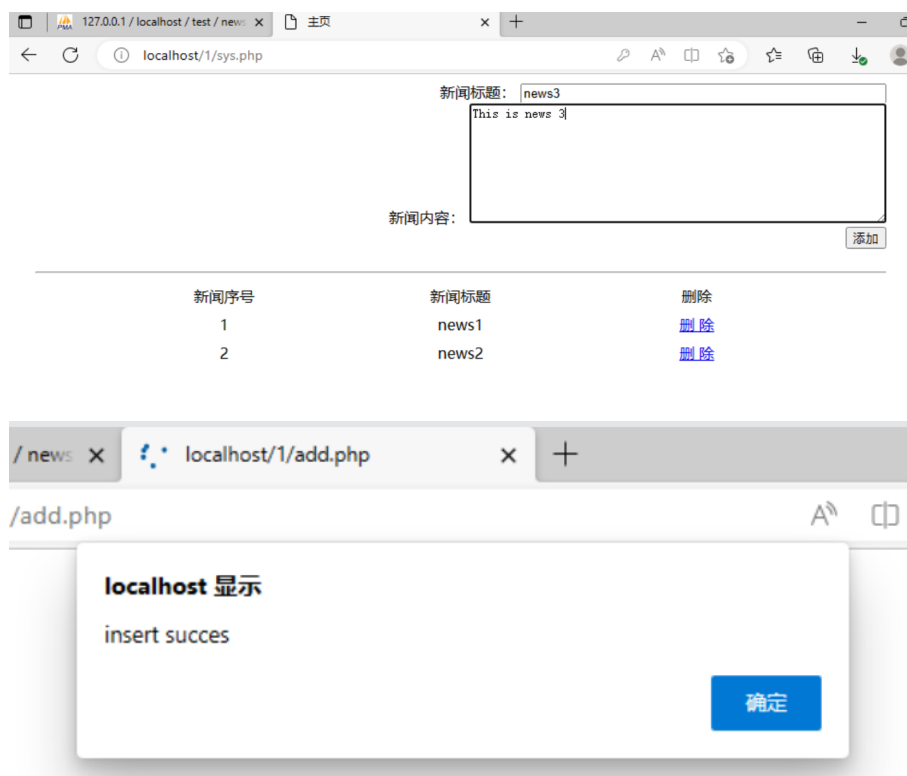
```

10 // 关闭连接
11 mysql_close($conn);
12 if ($result) {
13     ?>
14     <script>
15         alert("insert succes");
16         window.location.href = "sys.php";
17     </script>
18     <?php
19 } else {
20     ?>
21     <script>
22         alert("insert failed");
23         history.back();
24     </script>
25     <?php
26 }
27
28 ?>
29

```

首先建立数据库连接和选择数据库；随后，使用 `$_POST` 获取通过 HTTP POST 方法发送到脚本的名为“topic”和“content”的表单字段的值。然后，构建 INSERT 语句，插入到数据库表中。

查询是否插入成功。如果插入操作成功，会输出一个提示框，并重定向到“sys.php”页面；如果插入操作失败，会输出另一个提示框，并返回上一页。



3.4.2 新闻显示

在数据库中检索并显示新闻的代码片段如下：

```
1  <?php
2  $SQLStr = "select * from news";
3  $result = mysql_db_query("test", $SQLStr, $conn);
4  if ($row = mysql_fetch_array($result)) //通过循环读取数据内容
5  {
6      // 定位到第一条记录
7      mysql_data_seek($result, 0);
8      // 循环取出记录
9      while ($row = mysql_fetch_row($result)) {
10         ?>
11         <tr>
12             <td height="30">
13                 <div align="center">
14                     <?php echo $row[0] ?>
15                 </div>
16             </td>
17             <td width="400">
18                 <div align="center">
19                     <?php echo $row[1] ?>
20                 </div>
21             </td>
22             <td>
23                 <div align="center"><a href="del.php?newsid=<?php
24                 echo $row[0] ?> "> 删 除 </a>
25                 </div>
26             </td>
27         </tr>
28         <?php
29     }
30 }
31 ?>
```

使用 `mysql_fetch_row()` 函数循环获取每一行的数据。在循环过程中，代码会输出一个 HTML 表格的行，并在每一列中显示相应的新闻数据。

在表格的第一列中使用输出 `newsid`，在第二列中输出新闻的标题。

在第三列中使用 `<a>` 标签创建一个链接，链接到一个名为 `del.php` 的文件，并通过 URL 参数传递 `newsid`。

3.4.3 新闻删除

当点击删除时，链接到 del.php，进行删除。以下是代码片段：

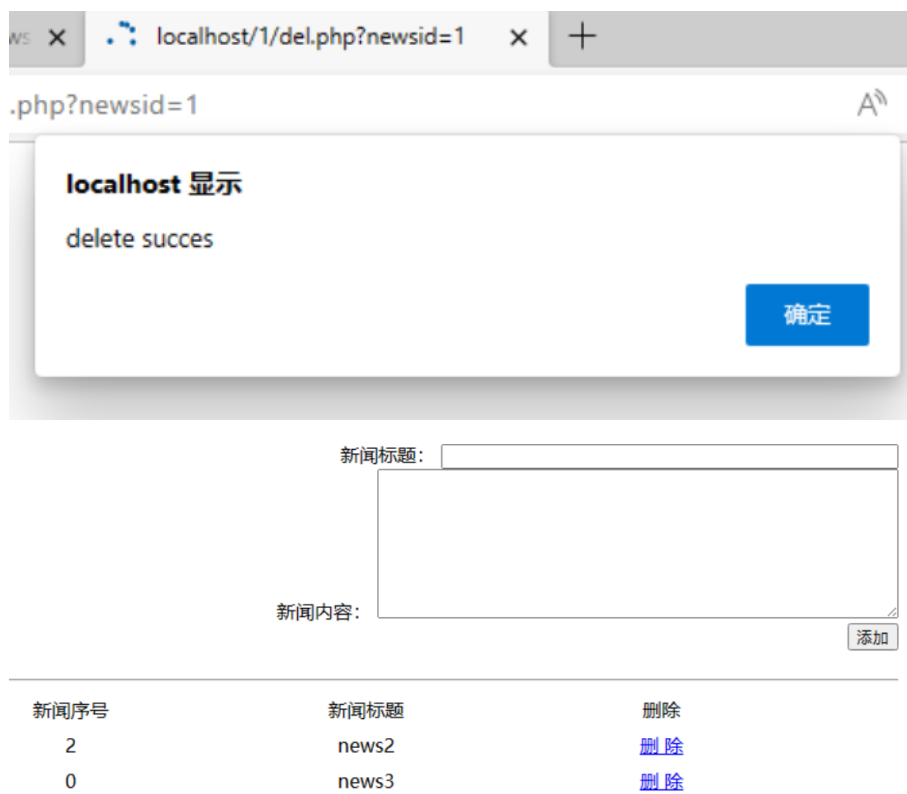
```
1  <?php
2  $conn = mysql_connect("localhost", "root", "root");
3  mysql_select_db("test");
4  $newsid = $_GET['newsid'];
5  $SQLStr = "delete from news where newsid=$newsid";
6  echo $SQLStr;
7  $result = mysql_query($SQLStr);
8  // 关闭连接
9  mysql_close($conn);
10 if ($result) {
11     ?>
12     <script>
13         alert("delete succes");
14         window.location.href = "sys.php";
15     </script>
16     <?php
17 } else {
18     ?>
19     <script>
20         alert("delete failed");
21         history.back();
22     </script>
23     <?php
24 }
25 ?>
26
```

首先建立与数据库的连接。从 `$_GET` 中获取名为“newsid”的参数值，该值通过 URL 查询字符串传递给脚本。

构建 SQL 语句，删除“news”表中的特定新闻记录，该记录的“newsid”列的值等于传递的“newsid”参数值。

如果删除成功，输出一个 JavaScript 弹窗显示“delete success”，并将浏览器重定向到“sys.php”页面。

如果删除失败，输出一个 JavaScript 弹窗显示“delete failed”，并将浏览器返回到前一页（通过 `history.back()` 函数）。



4 心得体会

通过本次实验，利用 php，编写了简单的数据库插入、查询和删除操作的示例，进一步了解 WEB 开发的细节。