

# 软件安全实验报告

## WEB开发实践

姓名：郭子涵 学号:2312145 班级：信息安全、法学双学位班

---

### 目录:

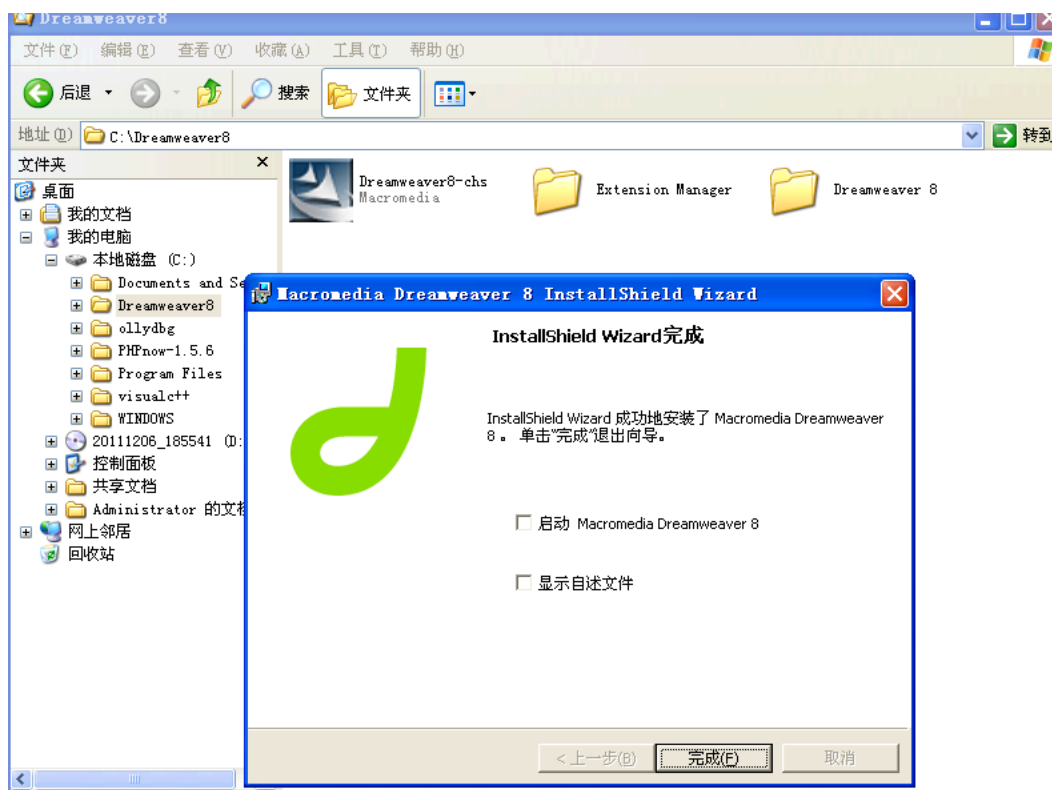
- 1 实验要求
- 2 实验内容
  - 2.1 安装Deamweaver8
  - 2.2 安装phpnow
  - 2.3 创建testDB数据库
  - 2.4 创建login.html
  - 2.5 编写login.php文件
  - 2.6 连接数据库并完成查询
  - 2.7 修改login.php
  - 2.8 创建news表
  - 2.9 创建sys.php
  - 2.10 创建addok.php, 实现nes表中的插入内容功能
  - 2.11 创建news.php, 创建表格显示news表中的id和topic字段
  - 2.12 创建see.php, 完善超链接输出内容
  - 2.13 创建del.php, 实现删除功能
- 3 心得体会

## 1 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。

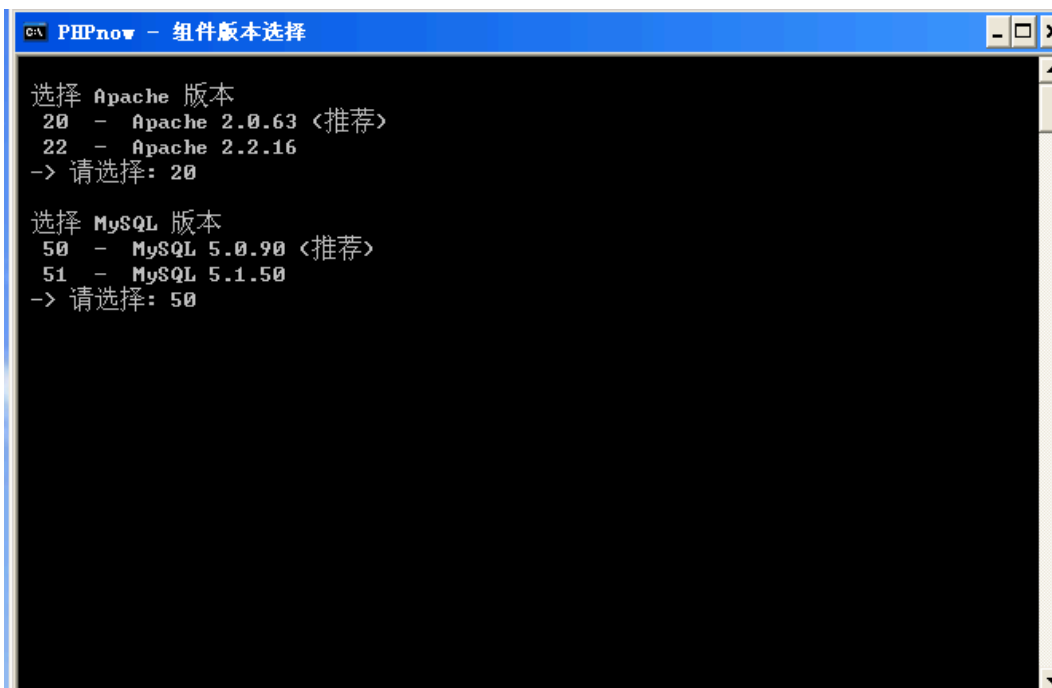
## 2 实验内容

### 2.1 安装Deamweaver8



### 2.2 安装phpnow

将资源中的压缩包放置在虚拟机中的xp系统下，解压后安装默认的Apache和MySQL版本：



```
Extracting Apache-20\icons
Extracting Apache-20\error\include
Extracting Apache-20\error
Extracting Apache-20\conf
Extracting Apache-20\cgi-bin
Extracting Apache-20\bin\iconv
Extracting Apache-20\bin
Extracting Apache-20

Everything is Ok

Folders: 120
Files: 2447
Size: 160548882
Compressed: 18827256

# 解压完成! 是否执行 Init.cmd 初始化?
-> 请选择(y/n): y^A
```

了解到xp后的操作系统自动提供管理员的机制，要求必须以指定管理员的身份才能安装，在本地中找到cmd.exe继续安装：

```
: 文件处理完成; :
: :
:
: 正在安装 Apache ... :
:
: 正在启动 Apache ... :
: 启动 Apache 完成; :
:
:
: 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: 123456
```

安装成功后，设置密码，跳转到网页127.0.0.1，实际访问的是默认index.php资源文件（解释性脚本，把php解释语句为对应的指令。

127.0.0.1

# Let's PHP now !

为何只能本地访问?

此服务器互联网 IP

60.29.153.12

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	2025-05-17 17:50:55 (+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	<input type="text" value="localhost"/>	MySQL 数据库名	<input type="text" value="test"/>
MySQL 用户名	<input type="text" value="root"/>	MySQL 用户密码	<input type="password"/>
			<input type="button" value="连接"/>

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

输入密码，测试数据库是否正常，显示OK没问题

127.0.0.1

# Let's PHP now !

为何只能本地访问?

此服务器互联网 IP  
60.29.153.12

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	2025-05-17 17:53:06 (+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	<input type="text" value="localhost"/>	MySQL 数据库名	<input type="text" value="test"/>
MySQL 用户名	<input type="text" value="root"/>	MySQL 用户密码	<input type="text"/>
			<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

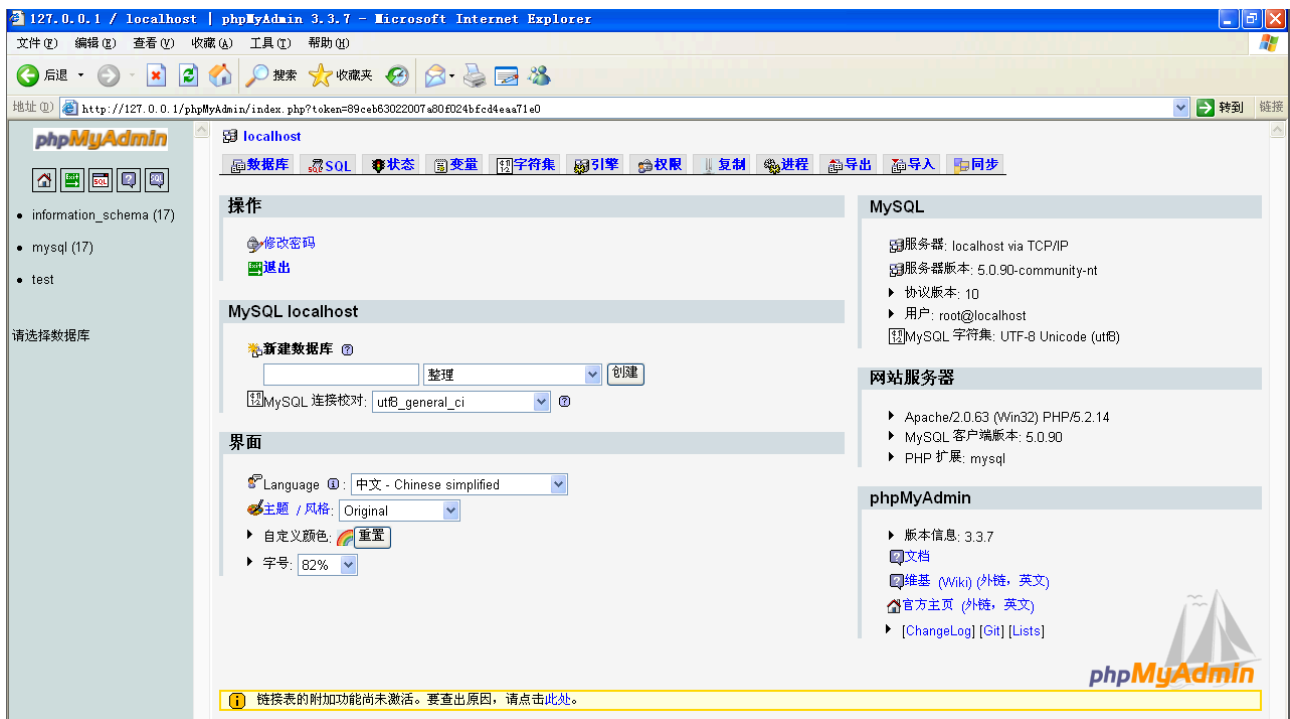
Valid XHTML 1.0 Strict / Copyleft ! 2007-? by PHPnow.org

点击进入phpMyAdmin,这是一个网页版的数据库管理系统

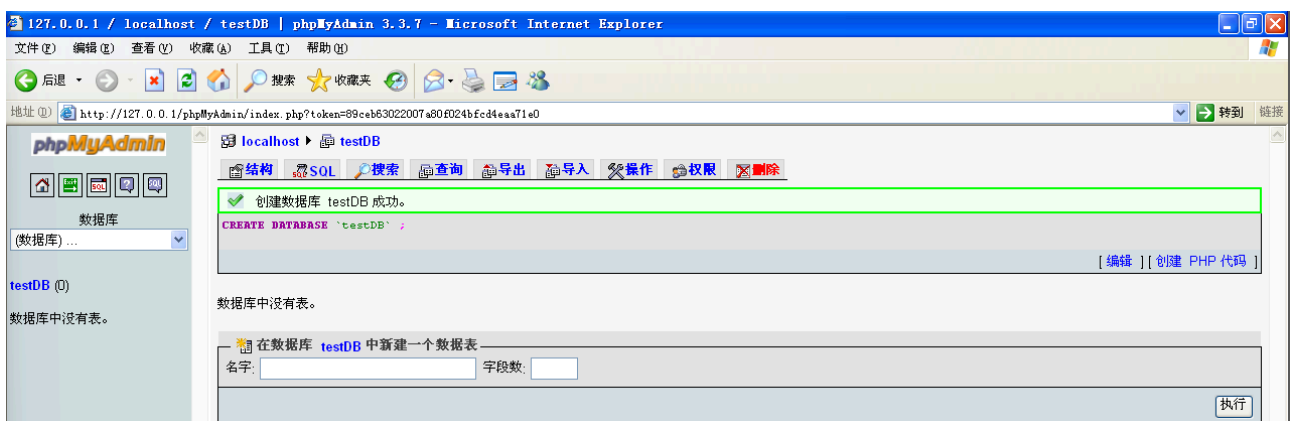


The image shows the phpMyAdmin login interface. At the top is the phpMyAdmin logo with a sailboat icon. Below it is the text '欢迎使用 phpMyAdmin'. There is a 'Language' dropdown menu set to '中文 - Chinese simplified'. Below that is a '登录' (Login) section with fields for '用户名:' (Username) and '密码:' (Password), and an '执行' (Execute) button. At the bottom, there is a yellow warning box with an information icon and the text '必须启用 Cookies 才能登录。' (Cookies must be enabled to log in).

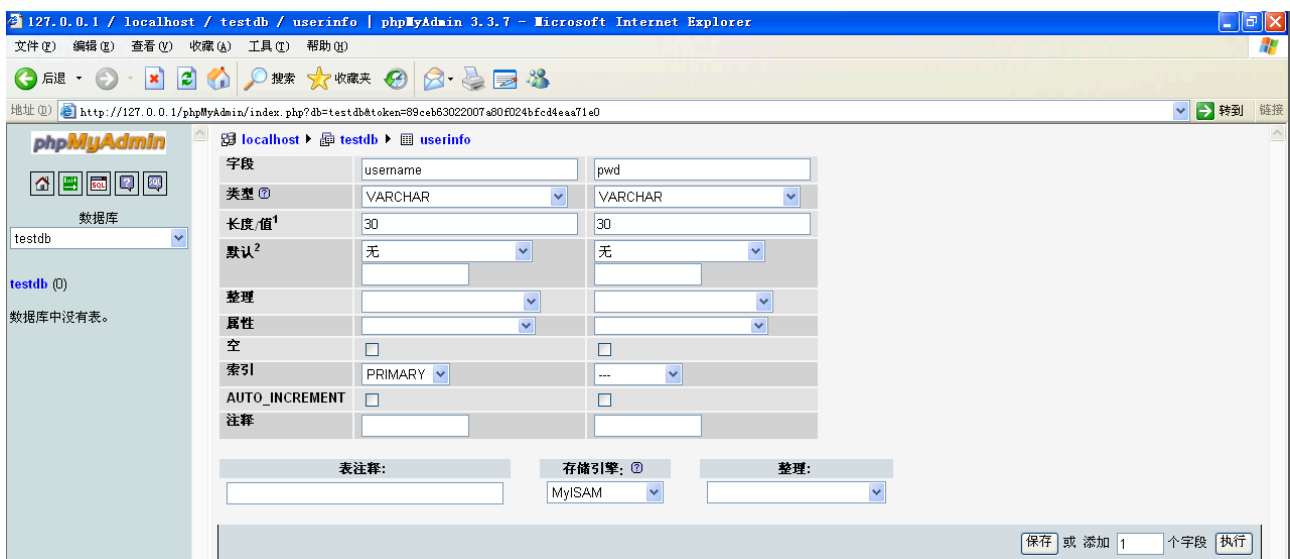
输入用户名和密码登录



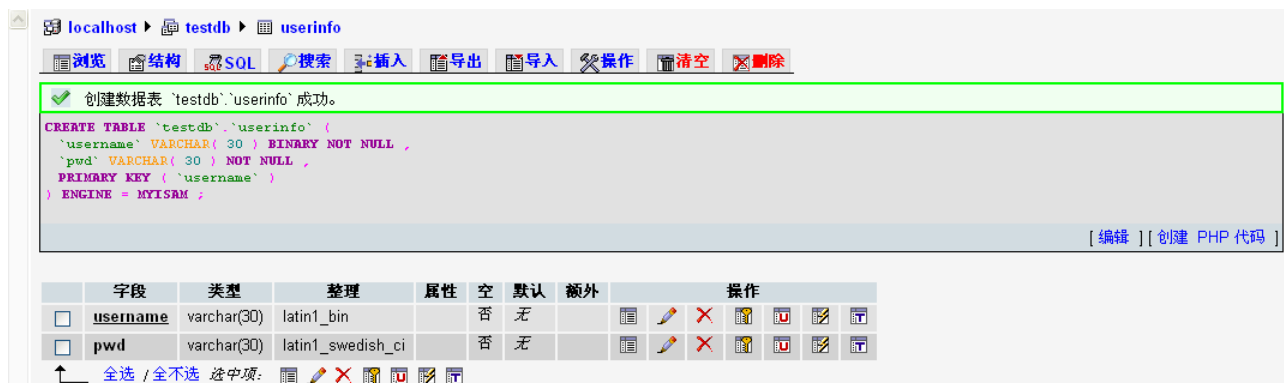
## 2.3 创建testDB数据库



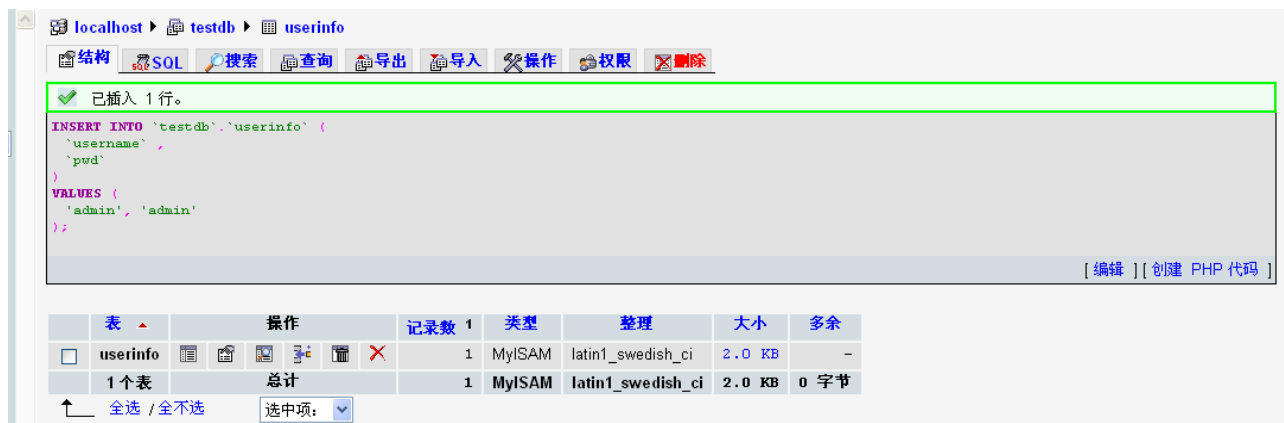
创建userinfo表，包含username和pwd两个字段



创建成功：



执行插入操作，插入用户名和密码均为'admin'的一条信息。

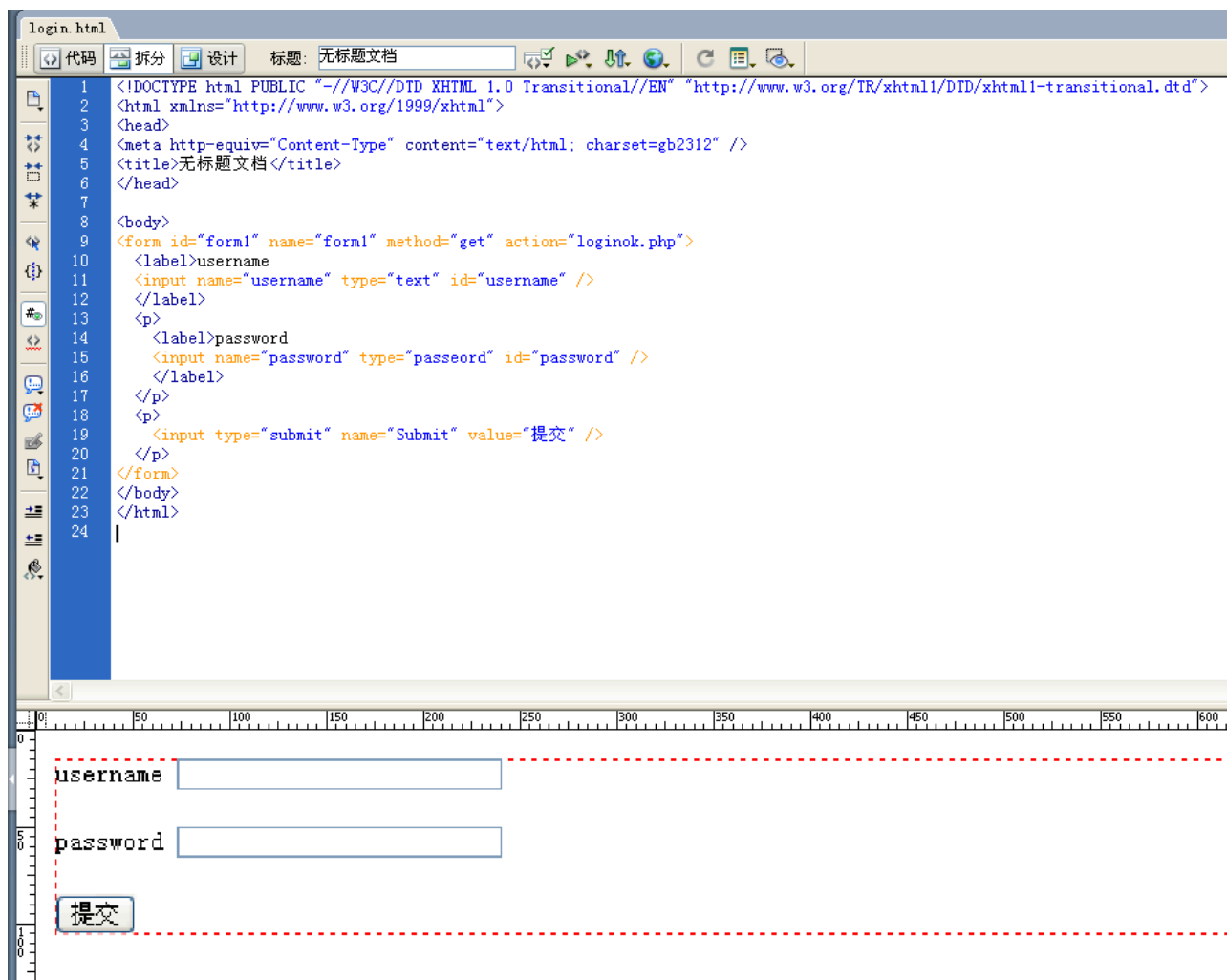


## 2.4 创建login.html

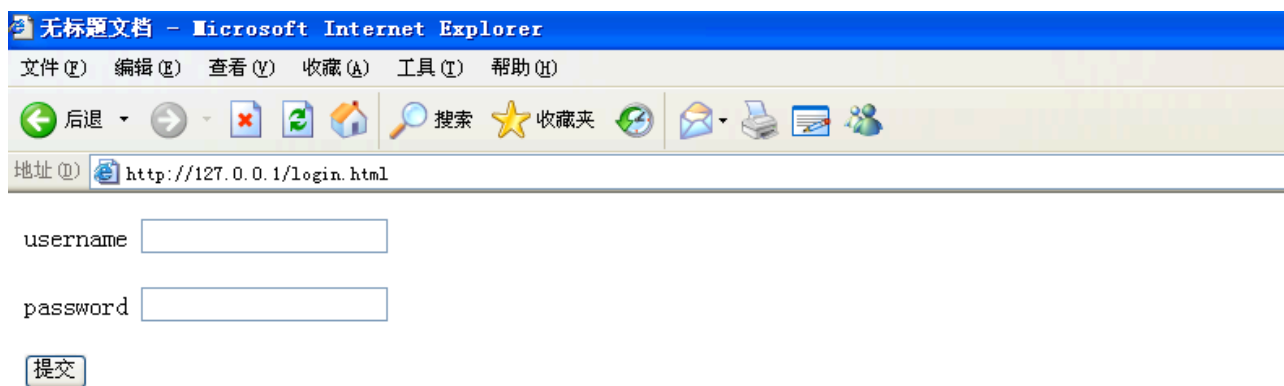
创建login.html,插入username和password两个文本域和提交按钮，为后续向表中插入数据做准备：

```
1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
4 <title>无标题文档</title>
5 </head>
6
7 <body>
8 <form id="form1" name="form1" method="post" action="loginok.php">
9   <label>username
10   <input name="username" type="text" id="username" />
11 </label>
12 <p>
13   <label>password
14   <input name="password" type="password" id="password" />
15 </label>
16 </p>
17 <p>
18   <input type="submit" name="Submit" value="提交" />
19 </p>
20 </form>
21 </body>
```

22 </html>

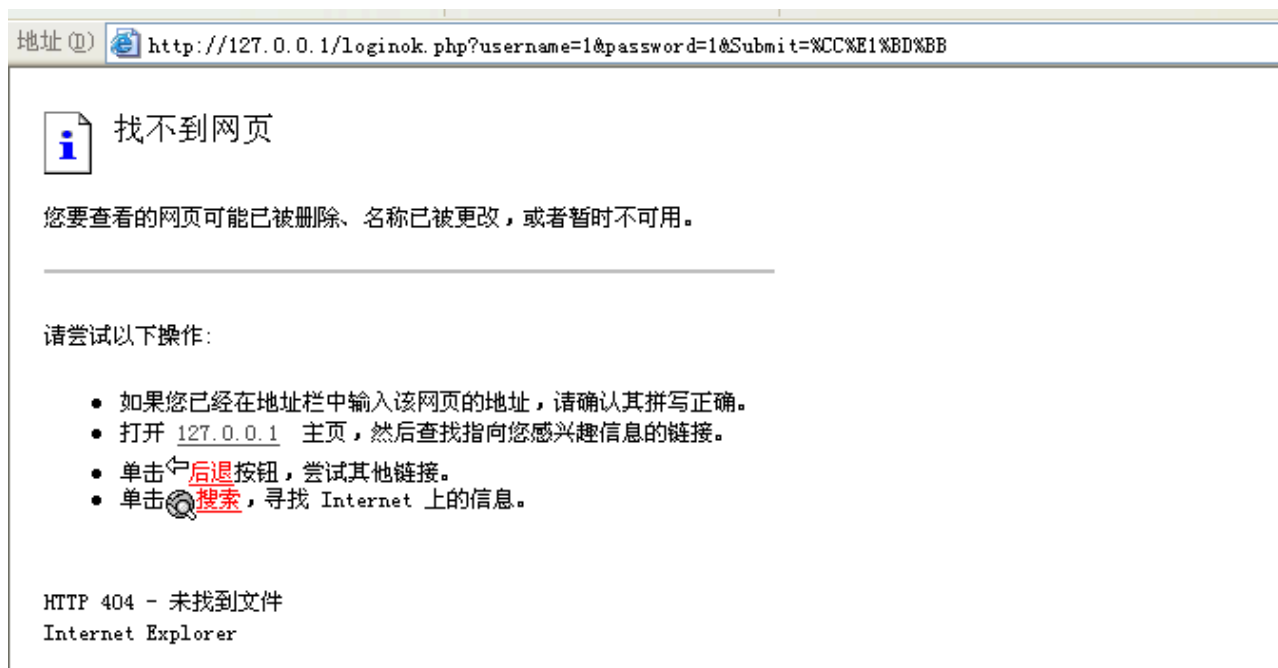


网页界面如下所示：



点击提交后触发form的提交事件，交给loginok.php脚本处理，这个脚本将获取用户名和密码的内容。由于请求方式为"get"，在url上可以看到请求的用户名和密码的内容都会在上面显示。



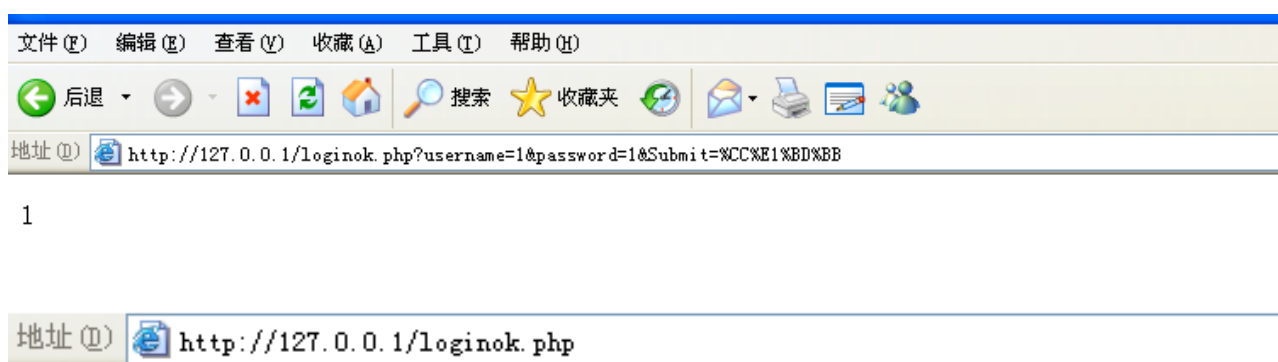


## 2.5 编写login.php文件

定义username和password两个变量，由于上述login.html定义使用get的方式获取，因此此处也用get的方式获取变量，然后打印username的值。

```
1 <?php
2 $username=$_GET['username']
3 $password=$_GET['password']
4 echo $username
5
6 ?>
```

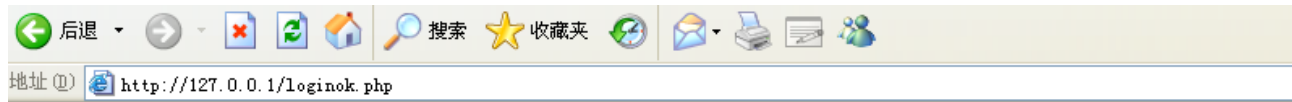
刷新后可以看到在网站上的打印出来的username



111222

## 2.6 连接数据库并完成查询

通过mysql\_connect连接数据库，mysql\_db\_query函数执行数据库SQL语句，mysql\_fetch\_array函数判断是否查找成功，如果能够成功去到，则返回为真，打印“ok”，否则返回为假。处理完之后，释放数据库。如果占用数据库连接没有释放，之后在此使用此脚本时，可能就没有办法完成连接的建立。



```
select * from userinfo where username='111222' and password='1'
```

```
1  <?php
2  $conn=mysql_connect("localhost", "root", "123456");    //连接数据库
3  $username = $_POST['username'];
4  $pwd = $_POST['password'];
5  $SQLStr = "SELECT * FROM userinfo where username='$username' and pwd='$pwd'";
6  echo $SQLStr ;
7  $result=mysql_db_query("testDB", $SQLStr, $conn); //执行数据库SQL语句
8  // 获取查询结果
9  if ($row=mysql_fetch_array($result))//读取数据内容
10     echo "<br>OK<br>";
11 else
12     echo "<br>false<br>";
13     // 释放资源
14     mysql_free_result($result);
15     // 关闭连接
16     mysql_close($conn);
17 ?>
```

输入错误和正确的用户名和密码验证：



```
SELECT * FROM userinfo where username='111222' and pwd='1'
false
```



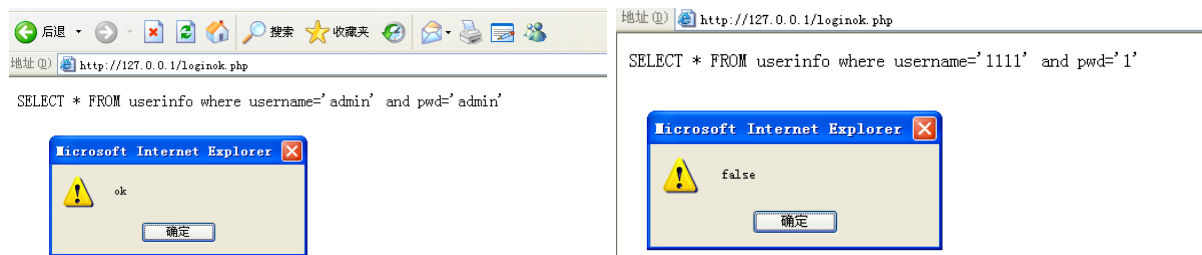
```
SELECT * FROM userinfo where username=' admin' and pwd=' admin'
OK
```

## 2.7 修改login.php

修改login.php实现当正确输入用户名和密码时，弹出ok窗口，错误输入时，弹出false窗口，并返回sys.php网页。

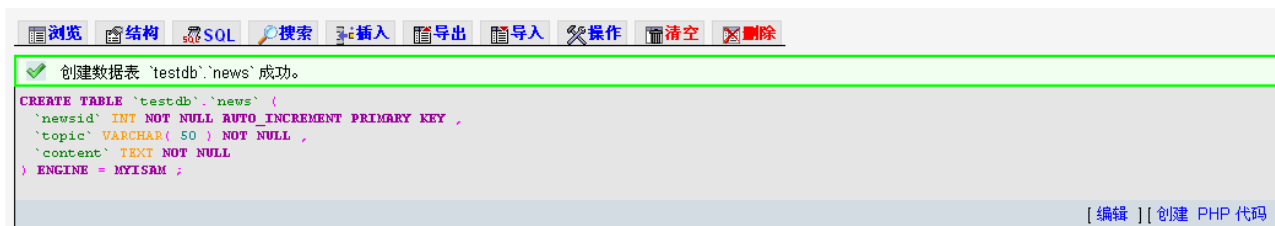
```
1  if ($isOK==1)
2  {
3      ?>
4      <script language="javascript">
5          alert("ok");
6          window.location.href="sys.php";
7      <?php
8  }else{
9      ?>
10     <script language="javascript">
11         alert("false");
12         history.bach();
13     }
14     ?>
```

执行验证，左图为正确输入结果，右图为错误输入结果：



## 2.8 创建news表

在phpMyAdmin中创建new表



## 2.9 创建sys.php

实现一个topic域，content文本区域和提交按钮，为news中实现插入操作布局：

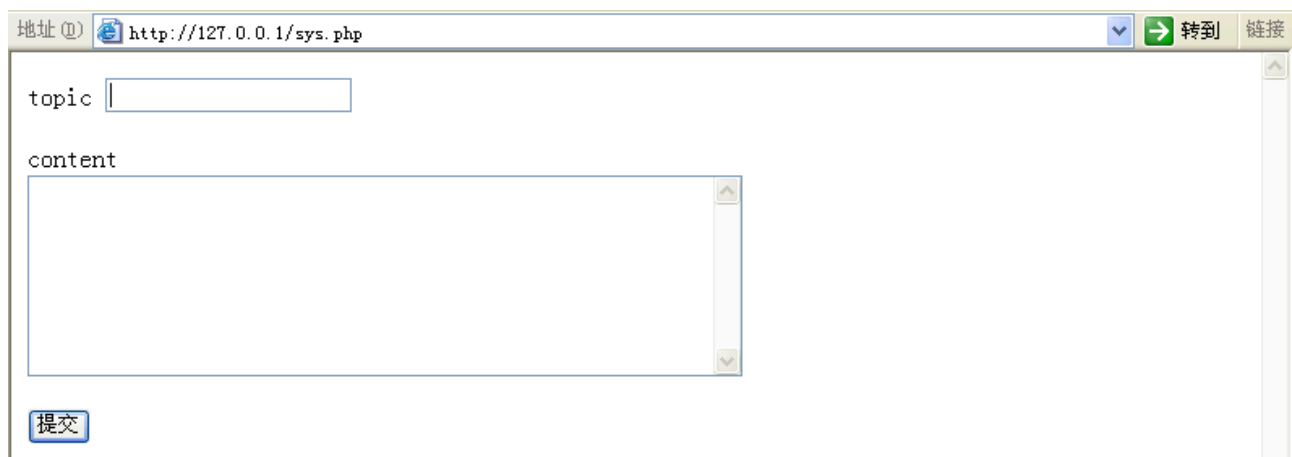
```
1  <html xmlns="http://www.w3.org/1999/xhtml">
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
4  <title>无标题文档</title>
```

```

5  </head>
6
7  <body>
8  <form id="form1" name="form1" method="post" action="addokok.php">
9      <label>topic
10     <input name="topic" type="text" id="topic" />
11     </label>
12     <p>
13         <label>content<br />
14         <textarea name="content" cols="60" rows="8" id="content"></textarea>
15         </label>
16     </p>
17     <p>&nbsp;</p>
18 </form>
19 </body>
20 </html>

```

运行验证：



The screenshot shows a web browser window with the address bar displaying 'http://127.0.0.1/sys.php'. The page contains a form with two input fields: 'topic' (a text box) and 'content' (a text area). Below the fields is a button labeled '提交' (Submit). The browser's address bar also shows a '转到' (Go) button and a '链接' (Link) button.

## 2.10 创建addok.php，实现nes表中的插入内容功能

实现向news表中插入内容功能：

```

1  <?php
2  $conn=mysql_connect("localhost", "root", "123456");    //连接数据库
3      mysql_select_db("testDB");
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

```

```

13  if ($result)
14  {
15      ?>
16      <script language="javascript">
17          alert("add ok");
18          window.location.href="sys.php";
19      </script>
20      <?php
21  }else{
22      ?>
23      <script language="javascript">
24          alert("add false");
25          history.bach();
26      </script>
27      <?php
28  }
29  ?>

```

在phpMyAdmin中可以看到表中确实实现了插入的数据：



## 2.11 创建news.php，创建表格显示news表中的id和topic字段

创建表格显示news表的两个字段。执行select语句，用my\_sql\_db\_query执行查询，用mysql\_fetch\_array函数判断是否读取数据内容，用mysql\_data\_seek定位到第一条记录，通过while循环，没去除一条记录，就输出。实现php脚本和html代码混合，组成复杂界面的显示

```

1  <html xmlns="http://www.w3.org/1999/xhtml">
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
4  <title>无标题文档</title>
5  </head>
6
7  <body>
8  <table width="600" border="1" align="center">
9      <tr>
10         <td>id</td>
11         <td>topic</td>

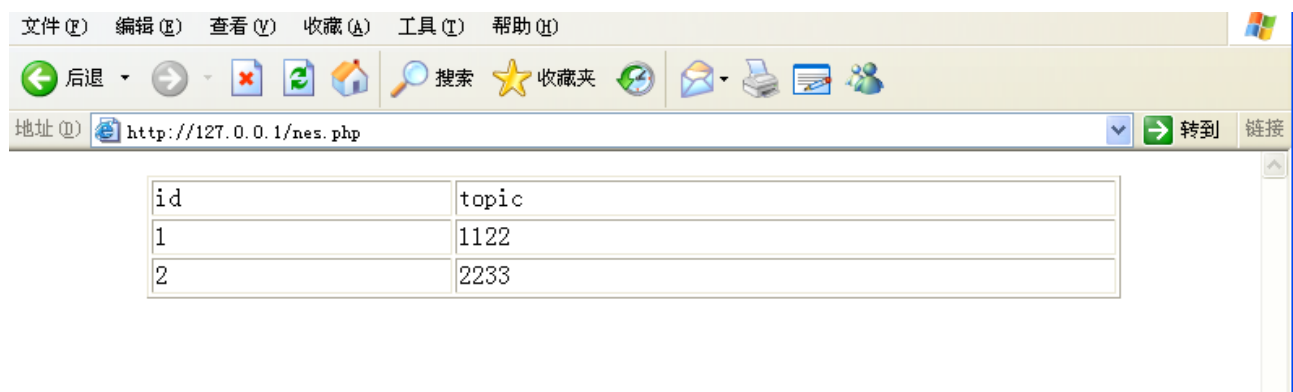
```

```

12     </tr>
13     <?php
14         $conn=mysql_connect("localhost","root","123456");
15         $SQLStr = "select * from news";
16         $result=mysql_db_query("testDB", $SQLStr, $conn);
17         if ($row=mysql_fetch_array($result))//通过循环读取数据内容
18         {
19             // 定位到第一条记录
20             mysql_data_seek($result, 0);
21             // 循环取出记录
22             while ($row=mysql_fetch_row($result))
23             {
24                 ?>
25                 <tr>
26                     <td>1</td>
27                     <td>1</td>
28                 </tr>
29             <?php
30                 }
31             }
32         ?>
33     </table>
34     <p>&nbsp;</p>
35 </body>
36 </html>

```

运行，显示结果：




修改代码，实现在第二个字段加上超链接：

```

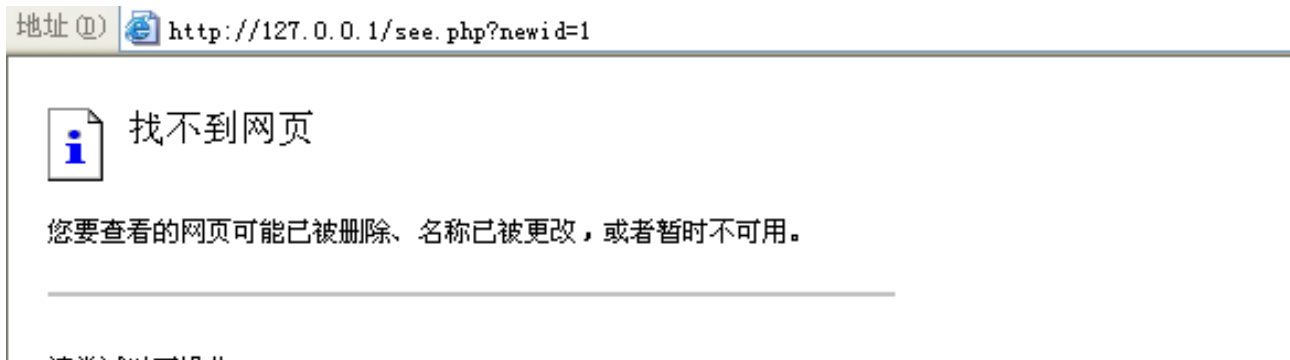
1     <tr>
2         <td><?php echo $row[0]; ?></td>
3         <td><a href="see.php?newid=<?php echo $row[0]; ?>"><?php echo $row[1]; ?>
4         </a></td>
5     </tr>

```

运行验证结果如下：

地址  http://127.0.0.1/nas.php 转到 链接

id	topic
1	<a href="#">1122</a>
2	<a href="#">2233</a>



## 2.12 创建see.php，完善超链接输出内容

see.php在new.php代码的基础上，保留数据库查询的操作,并输出记录的两个字段的内容

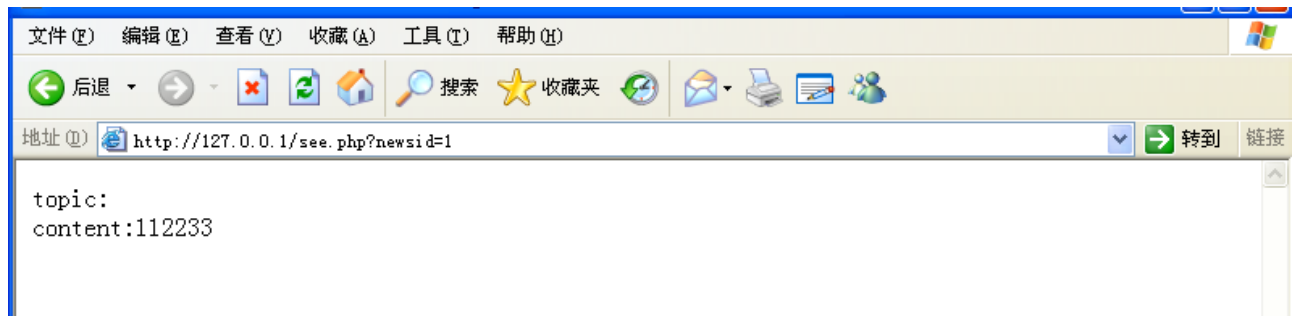
```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
5 <title>无标题文档</title>
6 </head>
7
8 <body>
9 <?php
10 $conn=mysql_connect("localhost","root","123456");
11 $id=$_GET['newid'];
12 $SQLStr = "select topic,content from news where newid='$id'";
13 $result=mysql_db_query("testDB", $SQLStr, $conn);
14 if ($row=mysql_fetch_array($result))//通过循环读取数据内容
15 {
16     // 定位到第一条记录
17     mysql_data_seek($result, 0);
18     // 循环取出记录
19     while ($row=mysql_fetch_row($result))
20     {
21         ?>
22         topic:<?php $row[0]; ?><br>
23         content:<?php echo $row[1]; ?><br>
24     <?php
25
26     }
```

```

27     }
28     ?>
29 </body>
30 </html>

```

刷新验证，最终实现内容正确输出：



## 2.13 创建del.php，实现删除功能

mysql\_query执行删除操作，然后返回删除成功与否的弹窗

```

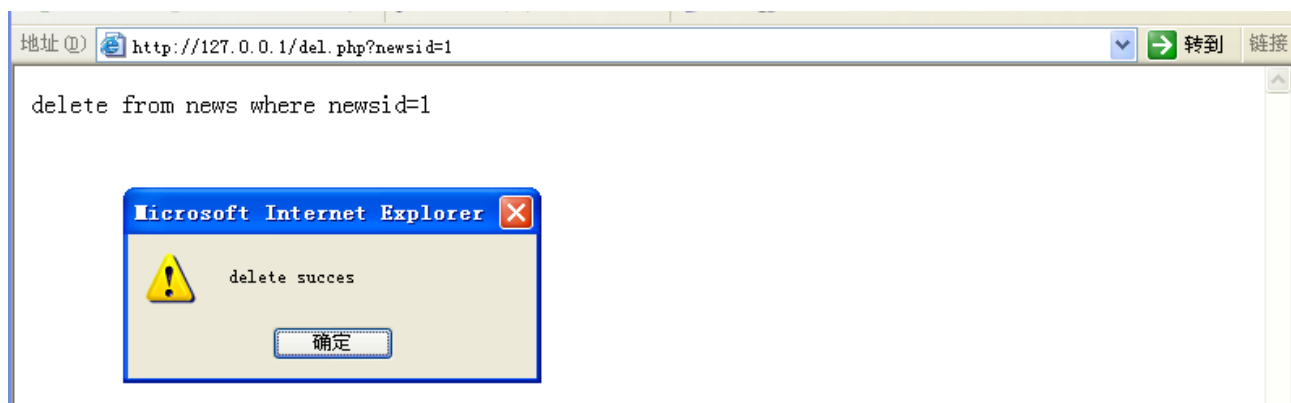
1  <?php
2  $conn=mysql_connect("localhost", "root", "123456");  mysql_select_db("testDB");

3  $newsid = $_GET['newsid'];
4  $SQLStr = "delete from news where newsid=$newsid";
5  echo $SQLStr;
6  $result=mysql_query($SQLStr);
7  // 关闭连接
8  mysql_close($conn);
9  if ($result)
10 {
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 ?>

```

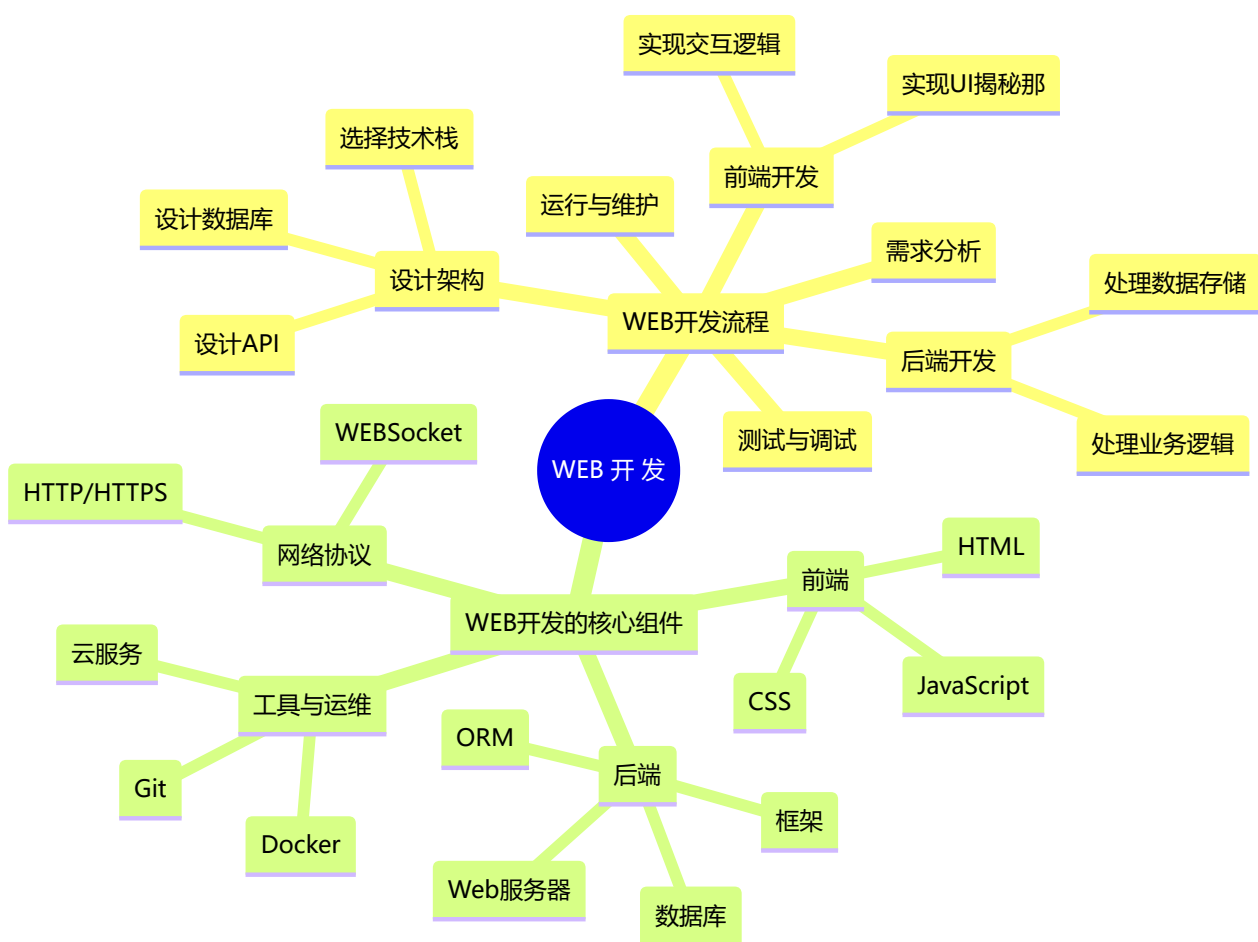
正确执行删除操作结果如下：





### 3 心得体会

本次实验进行了一个完整的案例，体会了用Dreamweaver开发自己的网站和数据库处理的完整流程。编写html和php脚本时也学习到php和html语言的语法，并与数据库连接，并对其中的表进行插入、增加、查询和删除等操作方面内容。对WEB开发有了初步的认识，总结其开发流程和核心组件如下思维导图所示：



同时本次实验在“post”和“get”两中不同的获取信息的差异中，更好地理解了前端与后端交互过程，对前端和后端的开发有了一定的了解，查阅资料了解到HTTP的其他常用请求方式：

#### 1、GET方法

- GET方法用于使用给定的URI从给定服务器中检索信息，即从指定资源中请求数据。使用GET方法的请求应该只是检索数据，并且不对数据产生其他影响。在GET请求的URL中发送查询字符串（名称/值对），需要这样写：  
`/test/demo_form.php?name1=value1&name2=value2`
- GET请求是可以缓存的，我们可以从浏览器历史记录中查找到GET请求，还可以把它收藏到书签中；且GET请求有长度限制，仅用于请求数据（不修改）。因GET请求的不安全性，在处理敏感数据时，绝不可以使用GET请求。

## 2、POST方法

POST方法用于将数据发送到服务器以创建或更新资源，它要求服务器确认请求中包含的内容作为由URI区分的Web资源的另一个下属。POST请求永远不会被缓存，且对数据长度没有限制；我们无法从浏览器历史记录中查找到POST请求。

## 3、HEAD方法

与GET方法相同，但没有响应体，仅传输状态行和标题部分。这对于恢复相应头部编写的元数据非常有用，而无需传输整个内容。

## 4、PUT方法

PUT方法用于将数据发送到服务器以创建或更新资源，它可以用上传的内容替换目标资源中的所有当前内容。它会将包含的元素放在所提供的URI下，如果URI指示的是当前资源，则会被改变。如果URI未指示当前资源，则服务器可以使用该URI创建资源。

## 5、DELETE方法

用来删除指定的资源，它会删除URI给出的目标资源的所有当前内容。

## 6、CONNECT方法

用来建立到给定URI标识的服务器的隧道；它通过简单的TCP /IP隧道更改请求连接，通常使用解码的HTTP代理来进行SSL编码的通信（HTTPS）。

## 7、OPTIONS方法

用来描述了目标资源的通信选项，会返回服务器支持预定义URL的HTTP策略。

## 8、TRACE方法

用于沿着目标资源的路径执行消息环回测试；它回应收到的请求，以便客户可以看到中间服务器进行了哪些（假设任何）进度或增量。

总体来说，本次实验收获颇丰。