《漏洞利用及渗透测试基础》实验报告

姓名： 齐明杰　　学号：2113997　　班级：信安 2 班

实验名称：

Web 开发实践

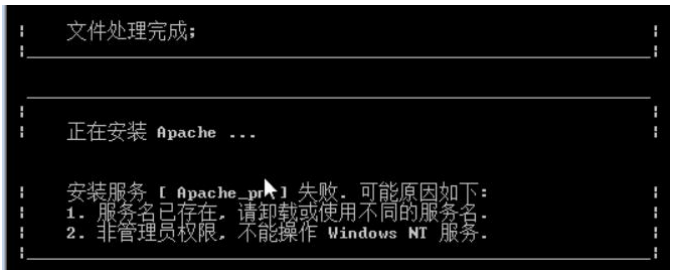
实验要求：

复现课本第十章实验三，利用 PHP，编写简单的数据库插入、查询和删除操作示例。基于课本的完整的例子，进一步了解 Web 开发细节。

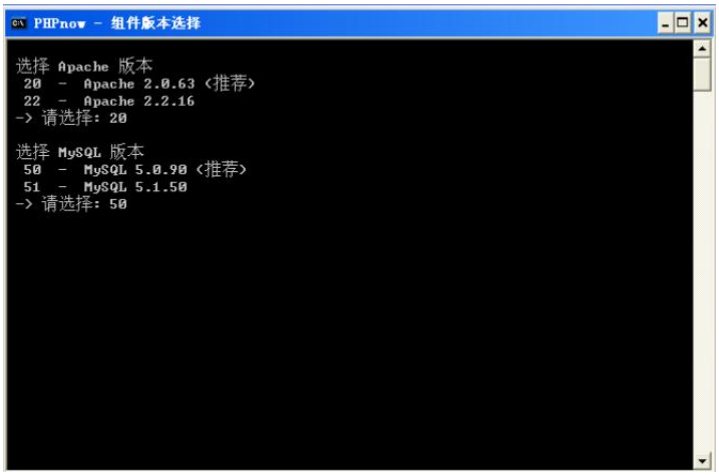
实验过程：

1、Phpnow 安装

由于 win7,win10 的管理员机制，不使用管理员权限运行 Init.cmd 会运行失败，如下图：

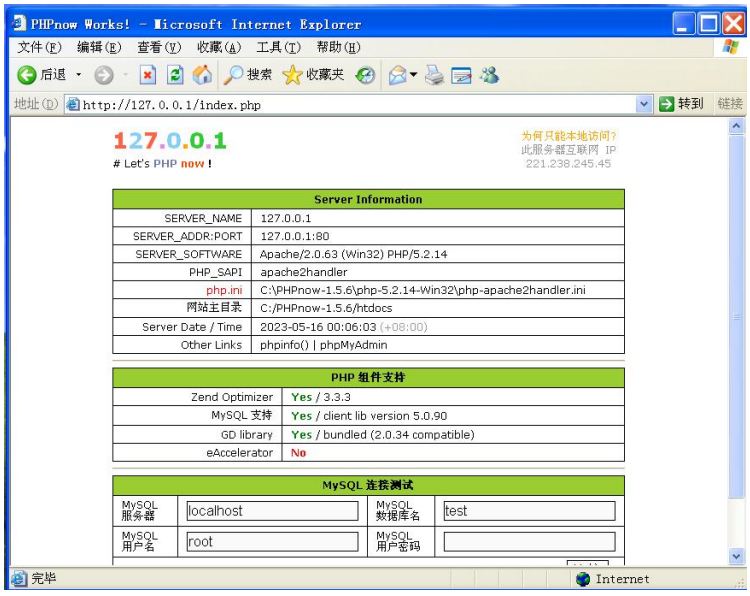


故我本次实验将采用 Windows XP 系统来进行实验。选择 Apache 和 Mysql 的版本:



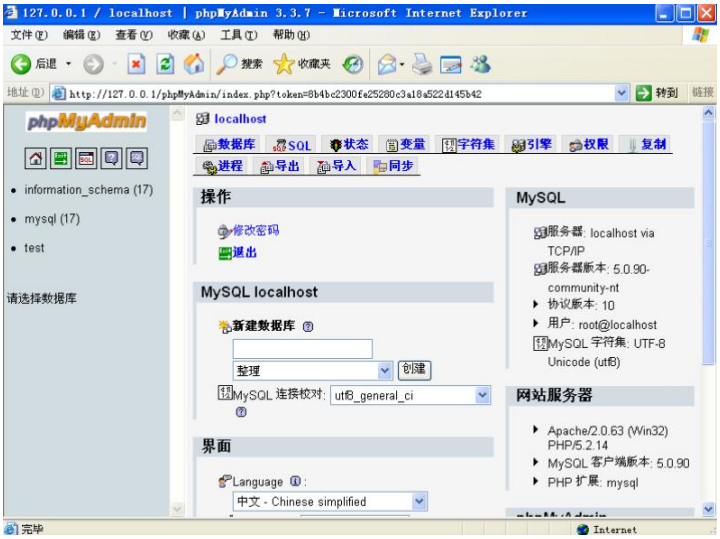
设置 MySQL 密码，完成安装

打开默认界面,即 http://127.0.0.1/index.php



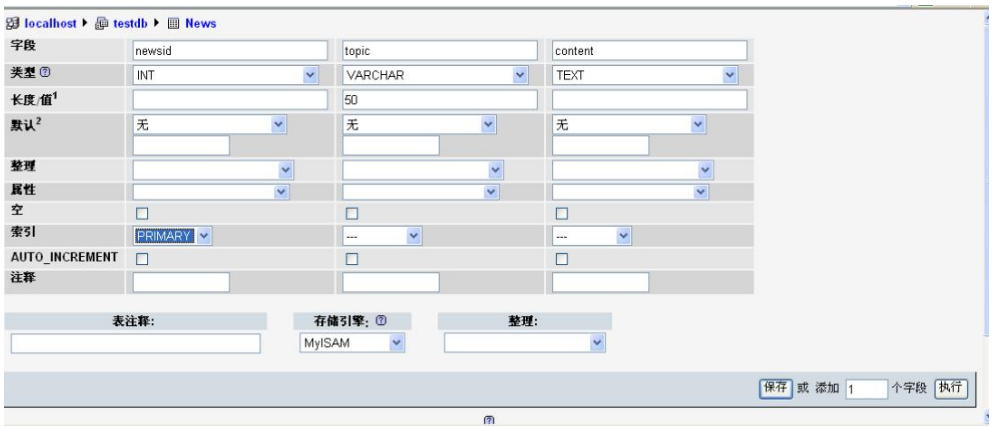
表示数据库成功连接

然后进入数据库管理系统

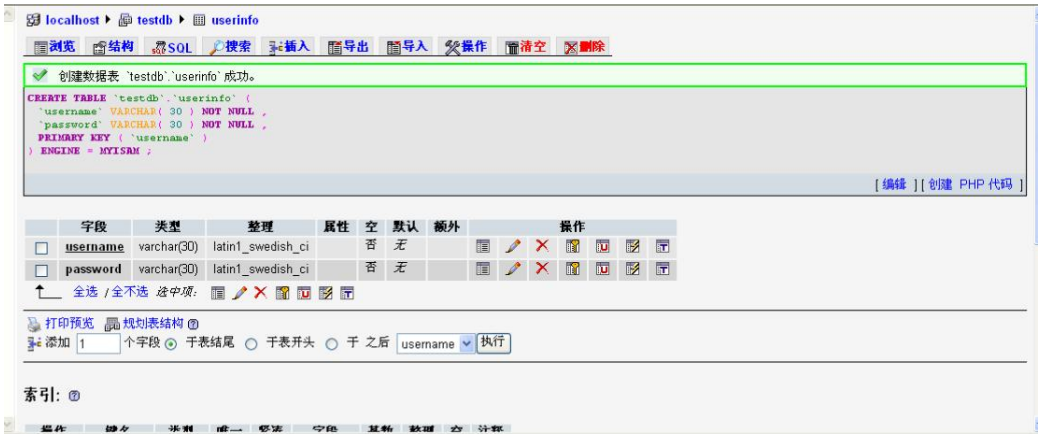


2、利用 php，编写简单的数据库插入、查询和删除操作的示例复现

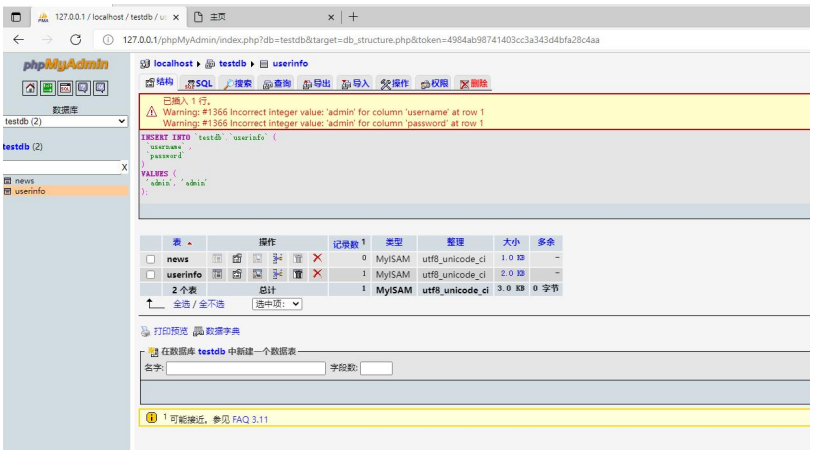
新建 TestDB 数据库，以及 News（newsid, topic, content）、userinfo（username, password）两张表,如下图所示:



创建成功



向表中插入数据：



然后新建 html 文件，准备写 html 代码，如下图:

新建 HTML 文件

3、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>
```

表单 <form>：这个表单有一个 ID 叫做 "form1"，并且使用 POST 方法向 "loginok.php" 发送数据。POST 是一种 HTTP 方法，用于向服务器发送数据。"loginok.php" 是处理这些数据的服务器端 PHP 文件。

输入 <input>：表单有两个输入字段。第一个输入字段的类型为 "text"，用于输入用户名。它的 name 和 id 都是 "username"。第二个输入字段的类型为 "password"，用于输入密码，它的 name 是 "pwd"，id 是 "pwd"。这两个字段都是必需的，用户需要在这里输入他们的用户名和密码。

按钮 <input type="submit">：这是一个提交按钮，用户点击它将会提交表单。按钮的值（即按钮上显示的文字）为 "提交"。

当用户填写了用户名和密码后，点击提交按钮，表单将会将这些数据以 POST 方法发送至 "loginok.php" 文件进行处理。

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
    }

?>
```

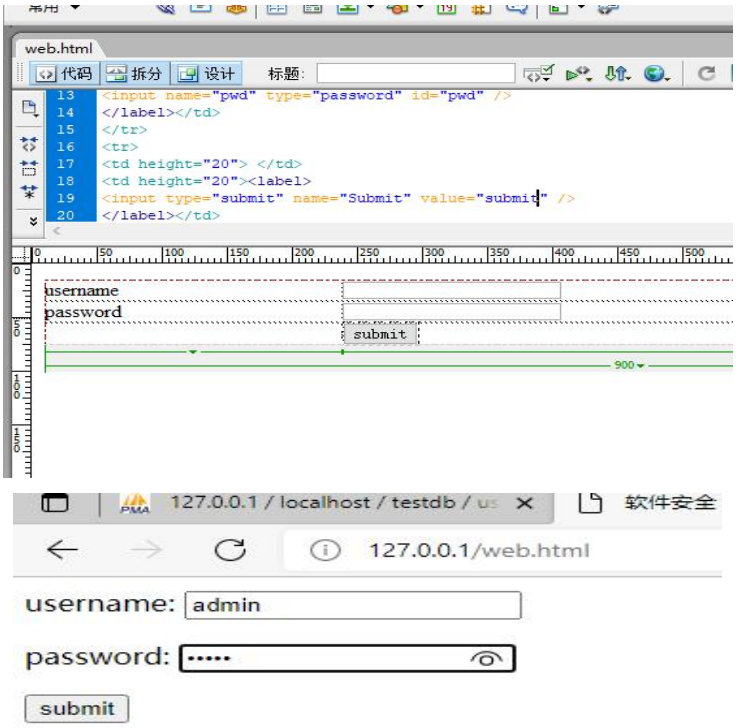
这段 PHP 代码的主要作用是处理用户登录。

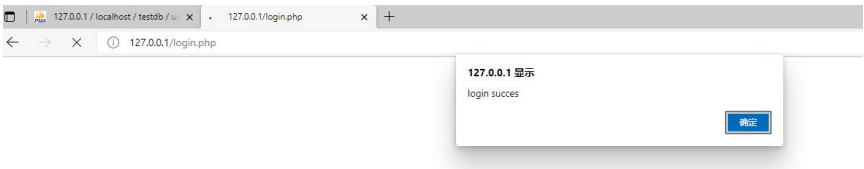
它首先尝试连接到本地的 MySQL 数据库，然后获取用户在 HTML 表单中提交的用户名和密码。然后，它在数据库的 userinfo 表中查找与这个用户名和密码匹配的记录。

如果找到了匹配的记录，那么它会设置一个标志变量 $loginok 为 1，表示登录成功。然后释放数据库查询的结果，关闭数据库连接。

接着，它会检查 $loginok 的值。如果为 1，就在网页上弹出一个提示框显示 "login success"，然后跳转到 "sys.php" 页面。如果 $loginok 不为 1，表示登录失败，就弹出一个提示框显示 "login failed"，然后返回到上一个页面。

运行效果如下:





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" cellspacing="0" cellpadding="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);
?>
```

代码的作用是连接到本地的 MySQL 数据库。

在页面的主体部分，首先创建了一个用于添加新闻的表单，该表单包含两个输入字段（新闻标题和新闻内容）和一个提交按钮。当用户填写新闻标题和内容后，点击"添加"按钮，将通过 POST 方法将数据发送到 "add.php" 文件进行处理。

之后，页面上有一个表格用于显示新闻列表。这个列表从数据库的 "news" 表中获取所有新闻记录，并将每一条新闻的序号，标题和一个删除链接显示出来。每个新闻的删除链接会指向 "del.php" 文件，并带上新闻的 id 作为参数。

代码的最后部分释放了数据库查询结果所占用的资源，并关闭了数据库连接。

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
    }

?>
```

首先，代码连接到本地的 MySQL 数据库，并选择 "testDB" 数据库。然后，从 POST 请求中获取 'topic' 和 'content' 字段的值，这些值是用户在 HTML 表单中提交的新闻标题和内容。

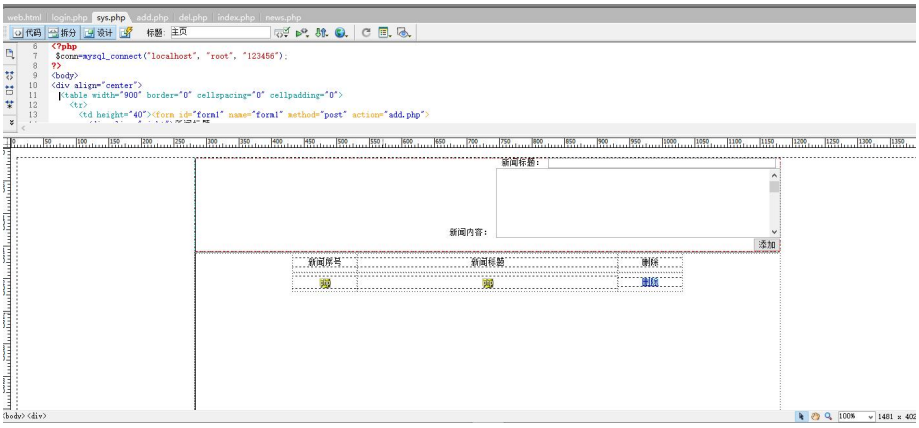
接下来，代码创建一个 SQL 插入语句，旨在将这个新闻的标题和内容插入到 "news" 表中，并执行这个 SQL 语句。

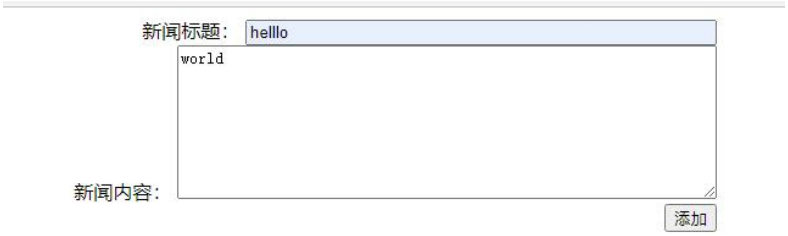
执行完 SQL 语句后，代码关闭了与数据库的连接。

最后一部分是根据 SQL 语句的执行结果决定下一步行动。如果 SQL 语句执行成功，代码将弹出一个显示 "insert success" 的提示框，然后页面会跳转到 "sys.php"。如果 SQL 语句执行失败，代码将弹出一个显示 "insert failed" 的提示框，并让页面返回到前一个页

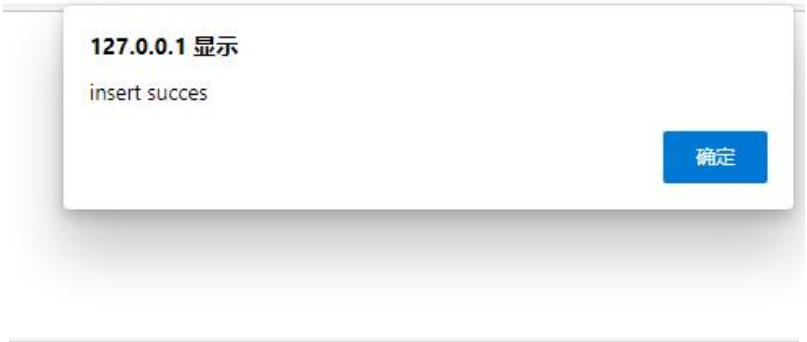
面。

执行效果：

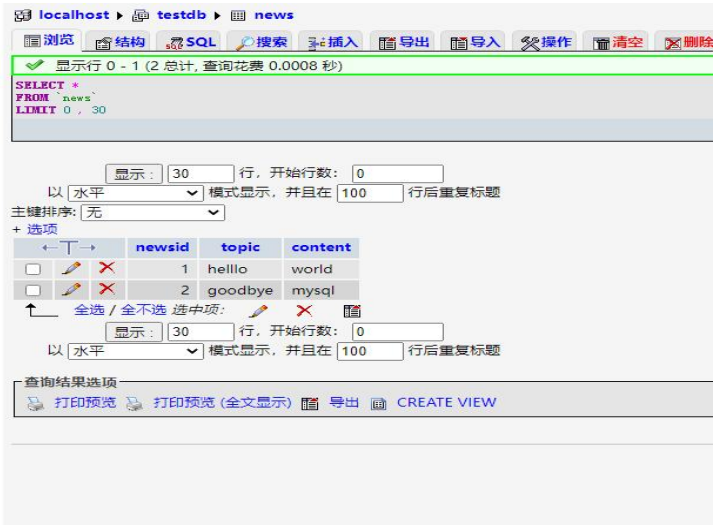




显示添加成功：



同时查看数据库内容，如下图所示：

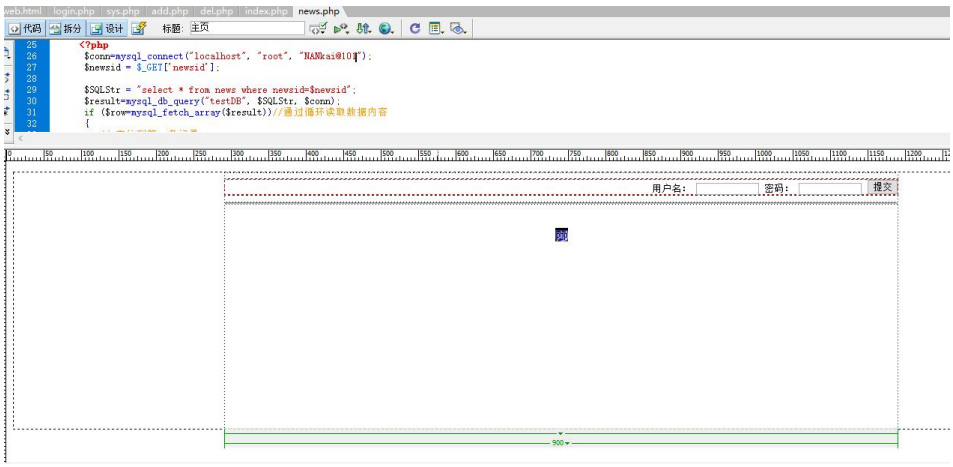


News.php：

```
<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>
```



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
        }
?>
```

首先，页面的顶部部分是一个用户登录表单，该表单包含两个字段：用户名和密码，以及一个提交按钮。当用户输入用户名和密码后，可以点击“提交”按钮，表单数据会被发送到 "loginok.php" 文件进行处理。

然后，页面的下方部分是用于显示新闻详情的。该部分的 PHP 代码首先连接到本地的 MySQL 数据库，然后从 GET 请求中获取 'newsid' 参数的值，这个参数值应该是用户在前一个页面点击新闻链接时传递过来的新闻 id。

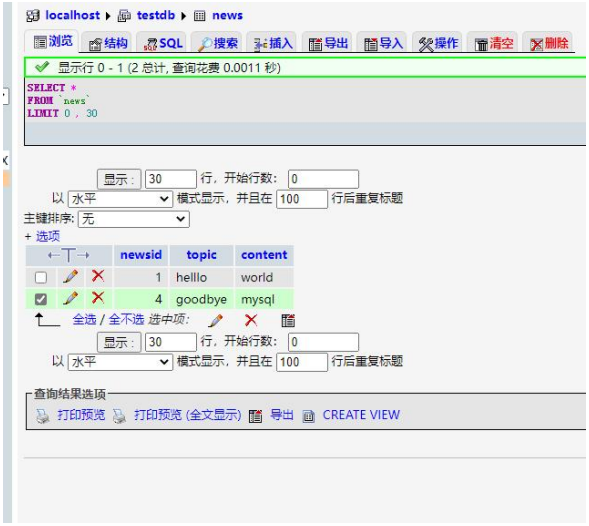
接着，代码根据这个新闻 id 创建一个 SQL 查询语句，用于从 "news" 表中获取该新闻的详细信息，然后执行这个 SQL 查询。如果查询结果中有数据，代码就会输出新闻的标题和内容。

最后，代码释放了查询结果所占用的资源，并关闭了数据库连接。

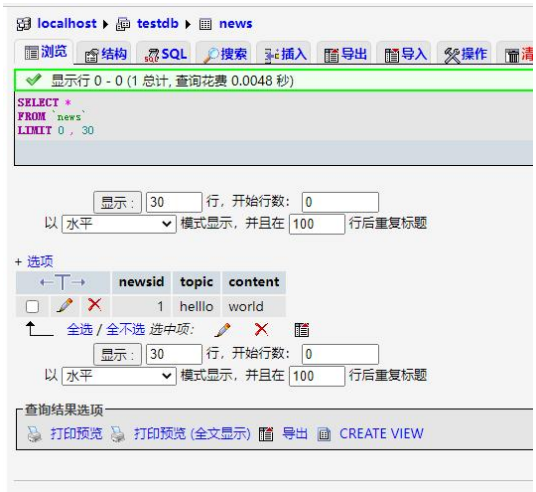
运行效果：

显示新闻序号和新闻标题，以及可以对某条新闻进行删除操作

删除第二条新闻前后数据库表内容对比：



删除成功



更新后 news.php：

```
<html>
<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="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"><table width="600" border="0" cellspacing="0" cellpadding="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>
        </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> <div align="center"> <a href="news.php?newsid=<?php echo $row[0] ?> " > <?php echo $row[1] ?>  </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);
?>
```

最后运行效果：

心得体会：

本次 Web 开发实验让我深入理解了 PHP 语言以及其与 HTML、SQL 数据库的交互机制。通过本次实验，我更好地理解了前端与后端的交互过程。我学会了如何在前端获取用户的输入，然后通过 POST 或 GET 方法发送到后端。在后端，我学会了如何接收这些输入，处理它们，并将结果发送回前端。

我学习并实践了如何通过 PHP 语言进行数据库的增删查改操作。虽然在这个过程中遇到了一些问题，但通过寻找资料和试验，我最终都成功解决了这些问题。