《软件安全》实验报告

姓名：蒋枘言　学号：2313546　班级：信息安全班

一、实验名称

Web 开发实践

二、实验要求

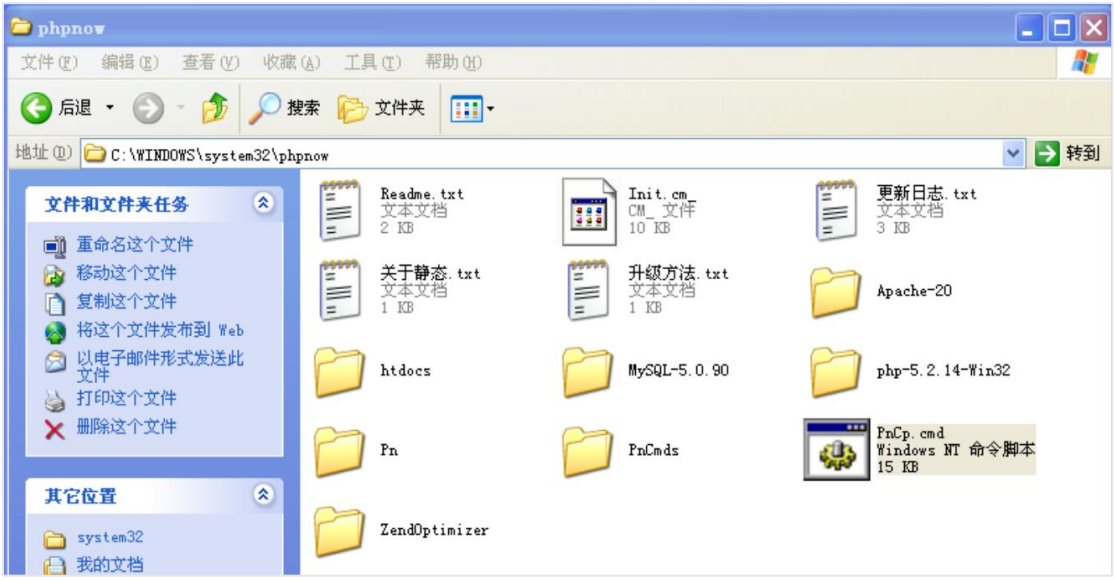
复现课本第十章的实验三（10.3.5 节）：利用 PHP，编写简单的数据库插入、查询和删除操作的示例。基于课本的完整的例子，进一步了解 Web 开发的细节。

三、实验过程

（一）安装 Dreamweaver 8 和 PHPnow

我们使用 Windows XP 虚拟机。Dreamweaver 8 的安装比较简单，但是 PHPnow 直接安装会安装失败，这时候需要以管理员身份运行 C:\WINDOWS\system32 里面的 cmd.exe，以命令行的方式安装。安装完成后设置 MySQL root 用户的密码。

安装完成的 phpnow 文件夹如图所示。

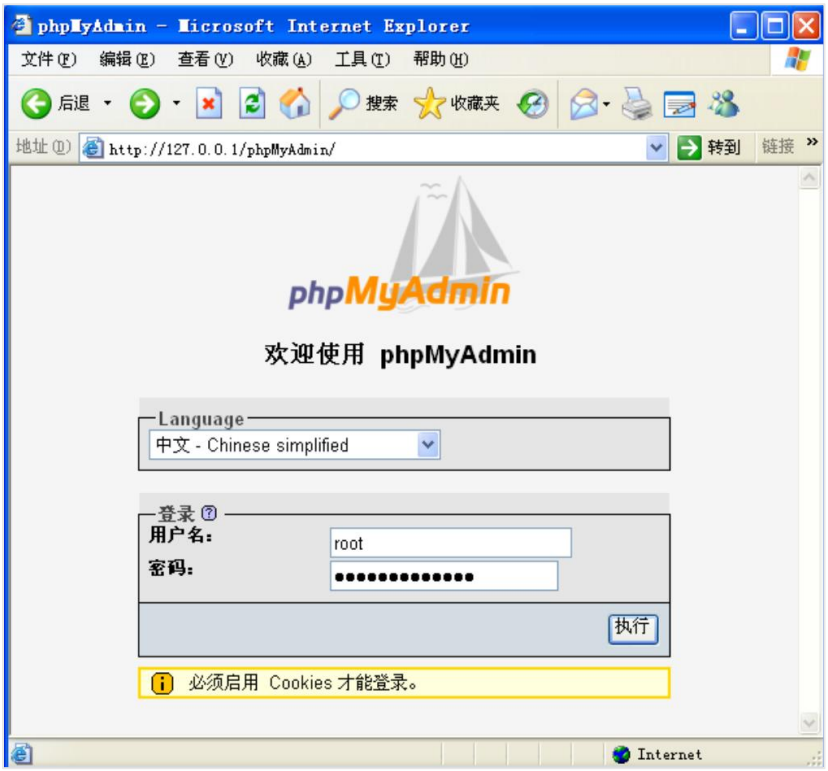


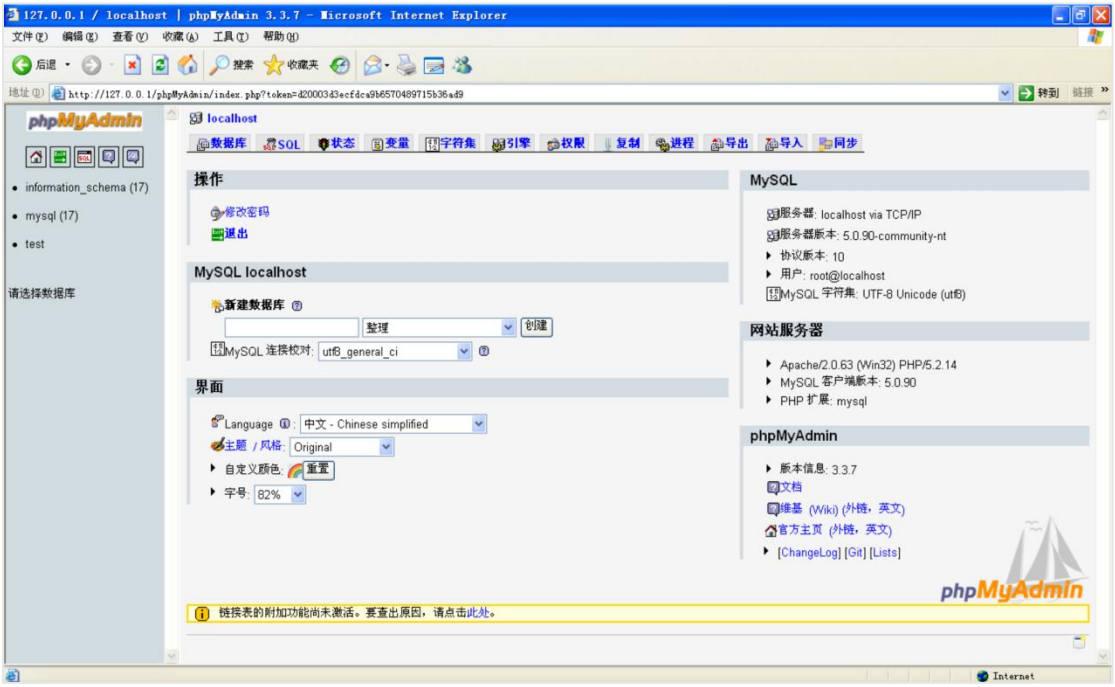
打开 PnCp.cmd，界面如下图所示，选择 20 即可启动 PHPnow。

（二）创建数据库

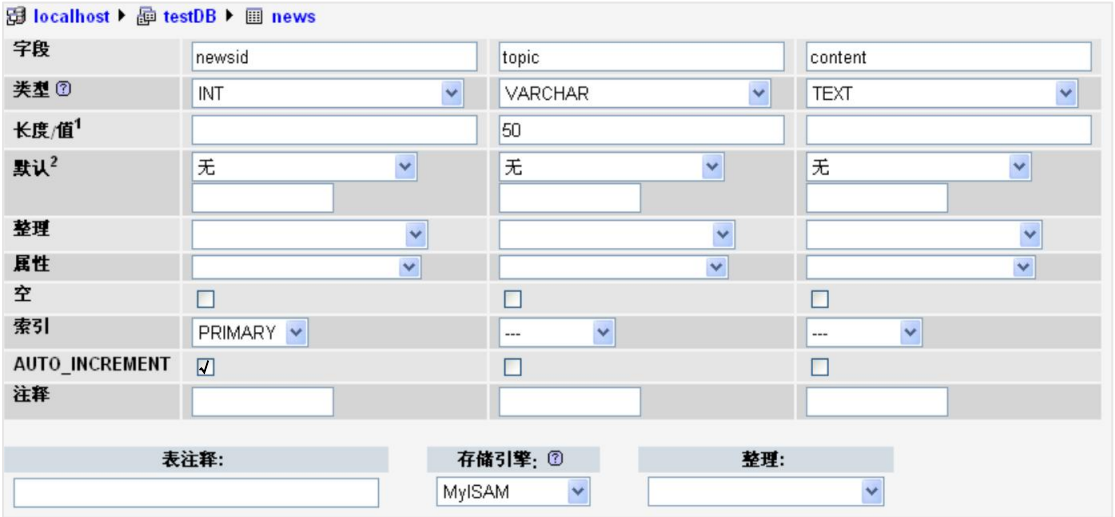
此时打开网页访问 http://127.0.0.1 就得到了如下图所示的界面：

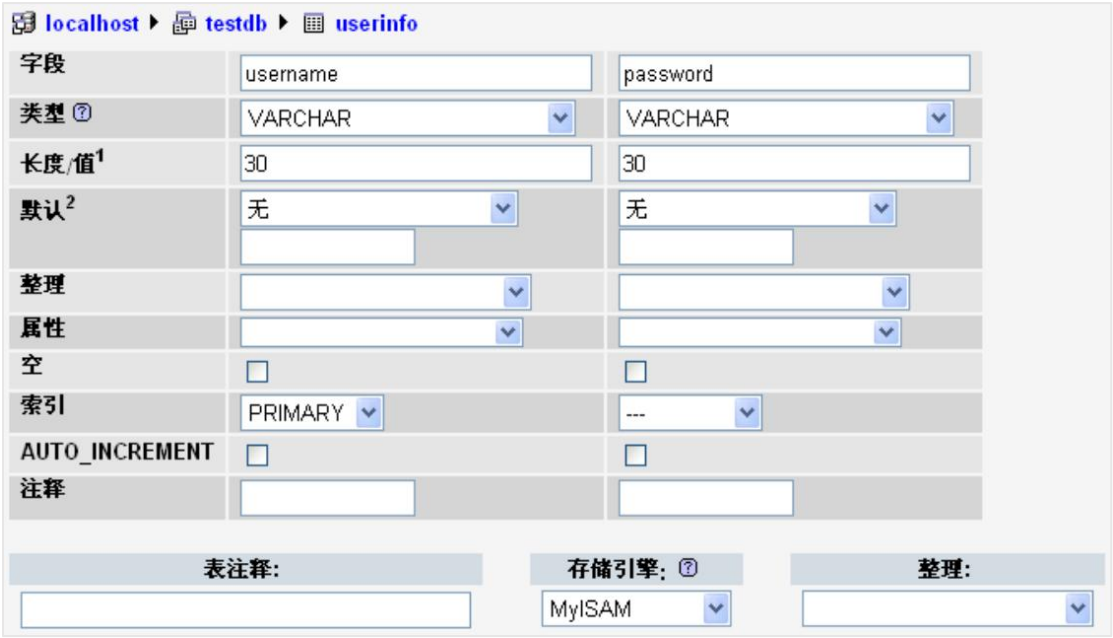
打开 phpMyAdmin，输入用户名和密码登录。





接下来创建数据库 testDB，包含两个表 **news**(newsid, topic, content)和 **userinfo**(username, password)。





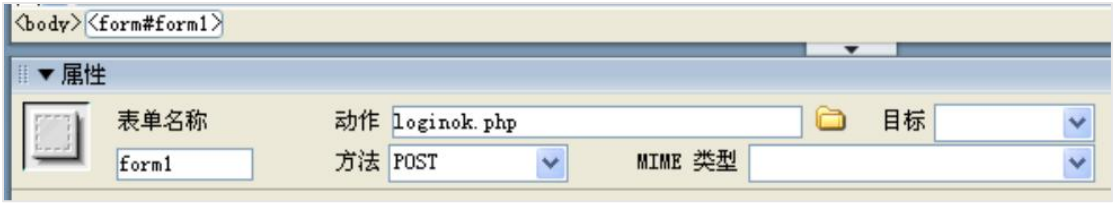
（三）编写 Web 程序

下面编写 PHP 文件。打开 Dreamweaver 8，新建 HTML 项目，输入以下代码，动作为 loginok.php，方法为 POST，并命名为 login.html，保存在 phpnow\htdocs 下。htdocs 是 PHPnow 的 Web 应用的根目录。

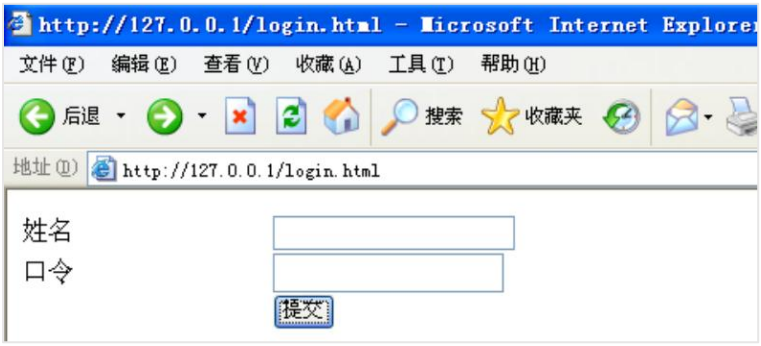
html

```
<!-- HTML 网页的开始 -->
<html>
<body>

<!-- 创建一个表单，id 和 name 都是 form1，使用 POST 方法提交到 loginok.php -->
<form id="form1" name="form1" method="post" action="loginok.php">

<!-- 创建一个 900 像素宽的表格，用于对齐输入框布局 -->
<table width="900" border="0" cellspacing="0" cellpadding="0">

  <!-- 第一行：输入姓名 -->
  <tr>
    <td height="20">姓名</td> <!-- 左边单元格显示标签“姓名” -->
    <td height="20">
      <label>
        <!-- 文本输入框，name 为 username，表单提交时这个名字会作为字段名 -->
        <input name="username" type="text" id="username" />
      </label>
    </td>
  </tr>

  <!-- 第二行：输入口令（密码） -->
  <tr>
    <td height="20">口令</td> <!-- 左边显示“口令” -->
    <td height="20">
      <label>
        <!-- 密码输入框，输入的内容会以星号/圆点显示 -->
        <input name="pwd" type="password" id="pwd" />
```

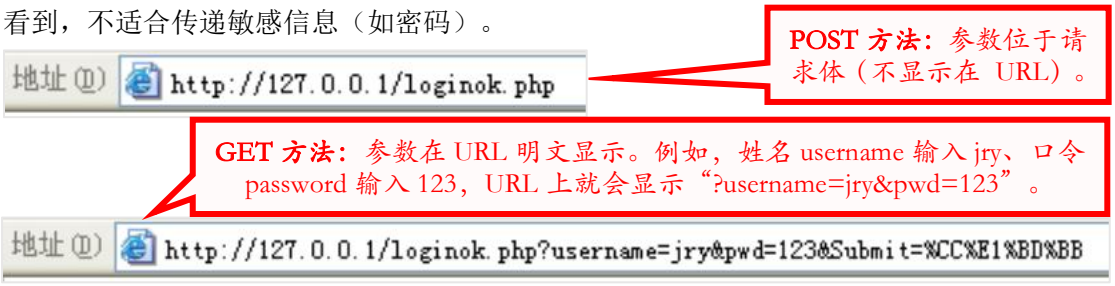
```
      </label>
    </td>
  </tr>

  <!-- 第三行：提交按钮 -->
  <tr>
    <td height="20"> </td> <!-- 左边空着，用于对齐 -->
    <td height="20">
      <label>
        <!-- 提交按钮，点击后会将表单中的 username 和 pwd 字段发送到
loginok.php -->
        <input type="submit" name="Submit" value="提交" />
      </label>
    </td>
  </tr>

</table>
</form>

</body>
</html>
```



打开 html 网页，得到如下界面，说明成功用 Dreamweaver 编辑得到了一个静态网页。



在这里 HTTP 与服务器交互的方法使用的是 POST，**POST 方法**主要用于向服务器**提交数据**，比如表单提交、文件上传、登录注册等，参数放在 **HTTP 请求体（body）**中，浏览器**不可见**，更加安全。而另一种 **GET 方法**主要用于从服务器**请求数据**，比如获取网页、查询数据，参数是放在 **URL 的“?”后面**，并且是**明文显示**的。由于参数是明文的，容易被看到，不适合传递敏感信息（如密码）。



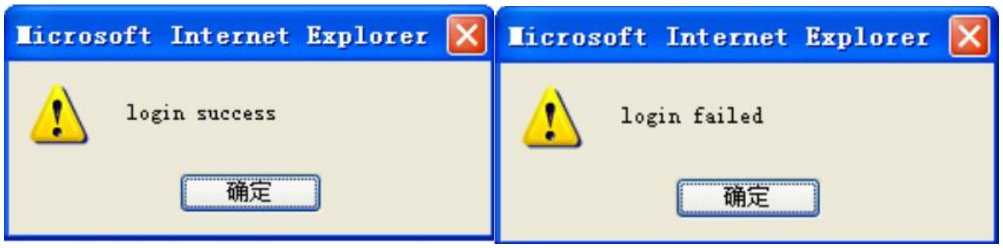
loginok.php（判断用户是否登录成功）代码的内容如下所示。

php

```php
<?php
// 初始化登录状态变量，默认为 0（表示登录失败）
$loginok = 0;

// 连接到 MySQL 数据库
$conn = mysql_connect("localhost", "root", "c 数据库密码 ");

// 获取用户提交的用户名和密码
$username = $_POST['username'];
$pwd = $_POST['pwd'];

// 构建 SQL 查询语句，验证用户名和密码是否匹配
$SQLStr = "SELECT * FROM userinfo WHERE username='$username' AND
password='$pwd'";

// 输出 SQL 查询语句（仅用于调试，部署时应删除）
echo $SQLStr;

// 执行查询语句，使用 testDB 数据库
$result = mysql_db_query("testDB", $SQLStr, $conn);

// 判断是否查询到用户记录（即验证通过）
if ($row = mysql_fetch_array($result)) {
    $loginok = 1; // 登录成功
}

// 释放结果资源，关闭数据库连接
mysql_free_result($result);
mysql_close($conn);

// 根据验证结果显示不同提示信息
if ($loginok == 1) {
    // 登录成功，跳转到系统页面
    ?>
    <script>
        alert("login success");
        window.location.href = "sys.php";
    </script>
    <?php
} else {
    // 登录失败，提示并返回上一页
    ?>
    <script>
        alert("login failed");
        history.back();
    </script>
    <?php
}
?>
```

首先在 userinfo 里面插入一个元组('jry', '123456')进行试验。若正确输入了用户名和密码

则弹出显示“login success”的对话框；若输入的用户名和密码不正确，则弹出显示“login failed”的对话框。



sys.php（新闻编辑）代码的内容如下所示。

php

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<!-- 设置网页编码为 gb2312（建议现代项目使用 UTF-8） -->
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>主页</title>
</head>

<?php
// 使用旧的 mysql 扩展连接数据库
$conn = mysql_connect("localhost", "root", "c 数据库密码 ");
?>

<body>
<div align="center">
<!-- 页面整体布局，宽度为 900 -->
<table width="900" border="0" cellspacing="0" cellpadding="0">
<!-- 添加新闻表单 -->
<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>

<!-- 分隔线 -->
<tr>
<td><hr /></td>
</tr>

<!-- 新闻列表展示 -->
<tr>
<td height="300" align="center" valign="top">
<table width="600" border="0" cellspacing="0" cellpadding="0">
<!-- 表头 -->
<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
// 查询所有新闻数据
$SQLStr = "SELECT * FROM news";
$result = mysql_db_query("testDB", $SQLStr, $conn);

if ($row = mysql_fetch_array($result)) {
    // 移动指针到第一条记录
    mysql_data_seek($result, 0);

    // 遍历所有记录
    while ($row = mysql_fetch_row($result)) {
        // $row[0] 是新闻 ID，$row[1] 是新闻标题
        ?>
        <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
// 释放资源并关闭数据库连接
mysql_free_result($result);
mysql_close($conn);
?>
```

add.php（添加新闻）代码的内容如下所示。

php

```
<?php
// 连接 MySQL 数据库服务器
$conn = mysql_connect("localhost", "root", "c 数据库密码 ");

// 检查连接是否成功
if (!$conn) {
    die("数据库连接失败: " . mysql_error());
}
```

```php
// 选择数据库 testDB 为当前操作对象
mysql_select_db("testDB", $conn);

// 从 POST 请求中获取新闻标题和内容（用户在表单中填写）
$topic = isset($_POST['topic']) ? htmlspecialchars($_POST['topic']) : '';
$content = isset($_POST['content']) ?
           htmlspecialchars($_POST['content']) : '';

// 构造插入 SQL 语句，将新闻标题和内容写入 news 表
$SQLStr = "INSERT INTO news (topic, content) VALUES ('$topic', '$content')";

// 输出 SQL 语句（仅用于调试）
echo $SQLStr;

// 执行插入操作
$result = mysql_query($SQLStr);

// 关闭数据库连接
mysql_close($conn);

// 判断插入是否成功，结果保存在 $result 中
if ($result) {
    ?>
    <!-- 如果插入成功，弹出提示并跳转到 sys.php 页面 -->
    <script>
        alert("Insert success");
        window.location.href = "sys.php";
    </script>
    <?php
} else {
    ?>
    <!-- 如果插入失败，弹出提示并返回上一页 -->
    <script>
        alert("Insert failed");
        history.back();
    </script>
    <?php
}
?>
```

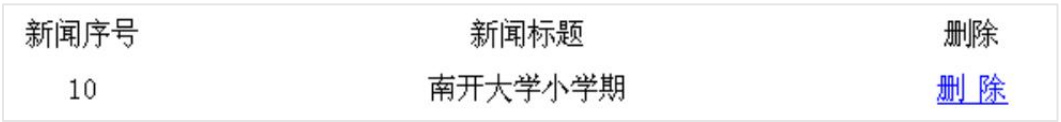
这样一来登录成功之后的界面如下图所示，可以添加新闻标题和新闻内容。

del.php（删除新闻）代码的内容如下所示。

php

```php
<?php
// 使用 mysql_connect 连接到本地数据库服务器
$conn = mysql_connect("localhost", "root", "c 数据库密码 ");

// 选择数据库 TestDB 作为当前操作数据库
mysql_select_db("TestDB", $conn);

// 从 URL 中获取 newsid 参数
$newsid = $_GET['newsid'];

// 构造 SQL 删除语句
$SQLStr = "DELETE FROM news WHERE newsid = $newsid";

// 输出 SQL 语句（调试用，正式部署时注释或移除）
echo $SQLStr;

// 执行 SQL 删除语句
$result = mysql_query($SQLStr);

// 关闭数据库连接
mysql_close($conn);

// 根据删除结果判断是否成功
if ($result) {
    ?>
    <script>
        alert("Delete success");
        window.location.href = "sys.php";
    </script>
    <?php
} else {
    ?>
    <script>
        alert("Delete failed");
        history.back();
    </script>
    <?php
}
?>
```

可以看到添加的每一条新闻后面都有删除按钮，用于删除。注意，删除后原来新闻的序号不会复用。



index.php（允许用户查看新闻和进行登录）代码的内容如下所示。

php

```php
<!DOCTYPE html>
<html>
<head>
    <meta charset="gb2312" />
    <title>主页</title>
```

```
</head>

<?php
// 连接数据库
$conn = mysql_connect("localhost", "root", "c 数据库密码 ");

// 检查连接是否成功
if (!$conn) {
    die("数据库连接失败: " . mysql_error());
}
?>

<body>
<div align="center">
<table width="900" border="0" cellspacing="0" cellpadding="0">
<!-- 登录表单 -->
<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="pwd" type="password" id="pwd" size="12" />
    <input type="submit" name="Submit" value="提交" />
</div>
</form>
</td>
</tr>

<tr>
<td><hr /></td>
</tr>

<!-- 新闻列表展示区域 -->
<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
// 查询所有新闻记录
$SQLStr = "SELECT * FROM news";

// 使用 mysql_db_query 查询指定数据库）
$result = mysql_db_query("testDB", $SQLStr, $conn);

// 如果查询有结果，则处理数据
if ($row = mysql_fetch_array($result)) {
    // 重置结果集指针到第一条记录
    mysql_data_seek($result, 0);

    // 循环读取每一行记录
    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
// 释放查询结果资源
mysql_free_result($result);

// 关闭数据库连接
mysql_close($conn);
?>
```

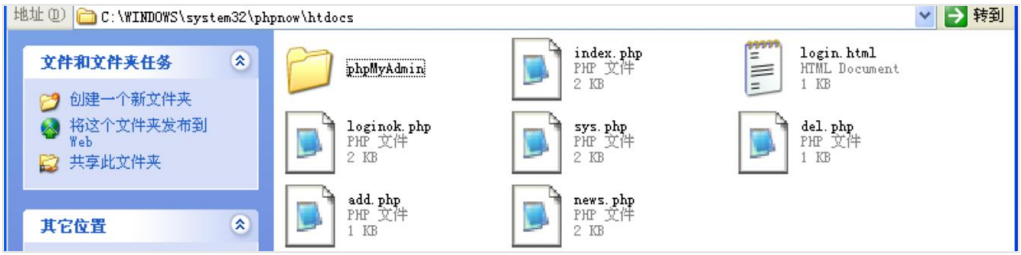
news.php（根据传入的 id 查看新闻内容）代码的内容如下所示。

php

```
<!DOCTYPE html>
<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">
<!-- 登录表单 -->
<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>
```

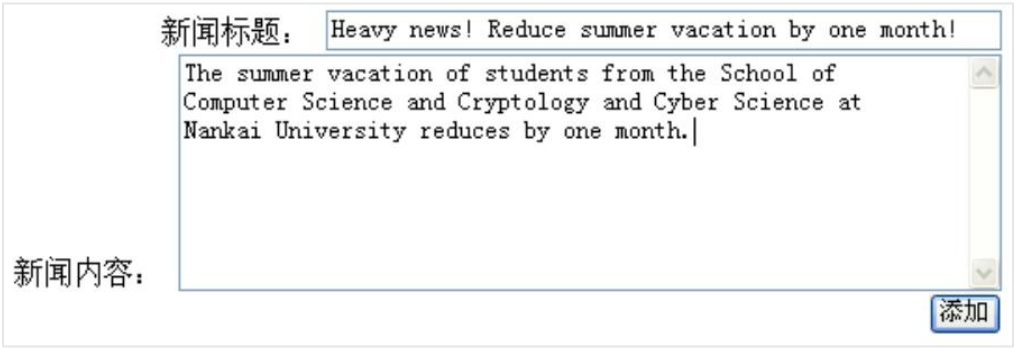
```
<tr>
<td><hr /></td>
</tr>

<!-- 新闻内容区域 -->
<tr>
<td height="300" align="center" valign="top">
<p> </p>
<?php
// 连接数据库
$conn = mysql_connect("localhost", "root", "c 数据库密码 ");

// 获取新闻 ID
$newsid = $_GET['newsid'];

// 构造 SQL 查询语句
$SQLStr = "SELECT * FROM news WHERE newsid = $newsid";

// 执行查询，返回结果集
$result = mysql_db_query("testDB", $SQLStr, $conn);

// 如果查询成功并有结果
if ($row = mysql_fetch_array($result)) {
    // 将指针移回第一条记录
    mysql_data_seek($result, 0);

    // 循环读取所有记录
    while ($row = mysql_fetch_row($result)) {
        // 输出新闻标题和内容
        echo $row[1] . "<br>";
        echo $row[2] . "<br>";
    }
}

// 释放查询结果
mysql_free_result($result);

// 关闭数据库连接
mysql_close($conn);
?>
</td>
</tr>
</table>
</div>
</body>
</html>
```

至此，一个简单的 Web 开发已经全部完成。

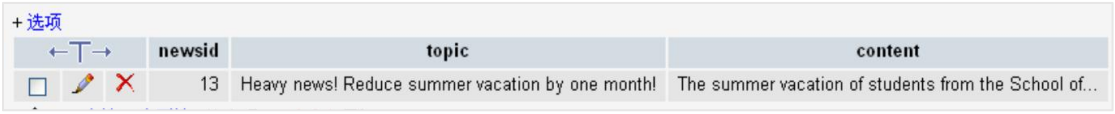


（四）可以开始使用编写好的 Web 应用

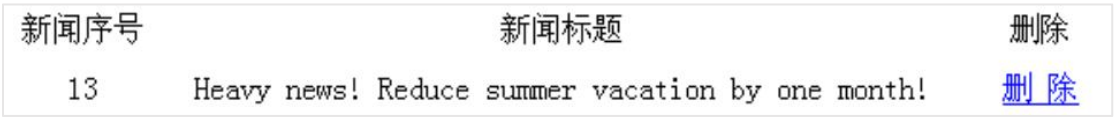
现在简单试用一下：输入正确的用户名和密码（口令），登录进去，进入可以增加和删除新闻的 sys.php 界面。我们增加一条新闻：

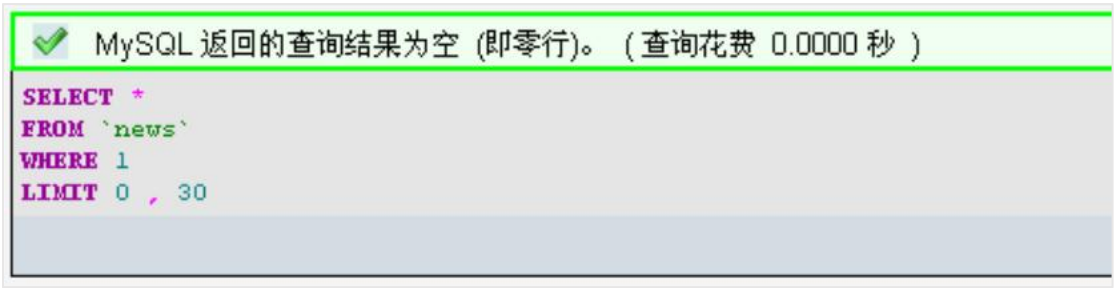


查看数据库中的数据，发现表 news 确实多出了这一条新闻。



删除后，这一条新闻在数据库中即被删除。





四、心得体会

这是我第一次接触到 HTML 和 PHP 语言，也第一次开发了一个 Web 应用程序。作为初学者，第一个 Web 应用程序也有许多改进的地方，比如：

1. 数据库连接改进

目前代码使用了 mysql_* 函数，这是一个已经废弃的 PHP 扩展，且存在安全隐患（如 SQL 注入）。可以使用更现代的 mysqli 或 PDO 扩展。

2. 字符编码优化

目前使用了 gb2312 字符编码，虽然对简体中文支持很好，但它相对较老，现代网页开发推荐使用 UTF-8 编码，因为它支持更多的字符集。

3. SQL 注入防范

当前的 SQL 查询方式直接将用户输入的内容拼接到 SQL 语句中，存在 SQL 注入风险。应使用预处理语句来防止 SQL 注入（以后会详细学到）。