《漏洞利用及渗透测试基础》实验报告

姓名：赵悦蛟 学号：2313650 班级： 1071

**实验名称：**

WEB开发实践

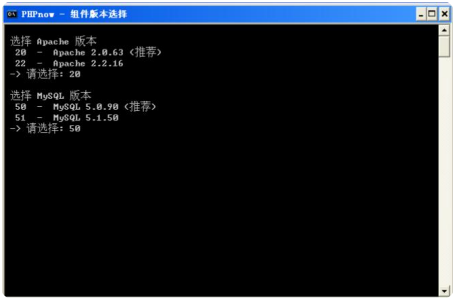
**实验要求：**

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。基于课本的完整的例子，进一步了解WEB开发的细节。

**实验过程：**

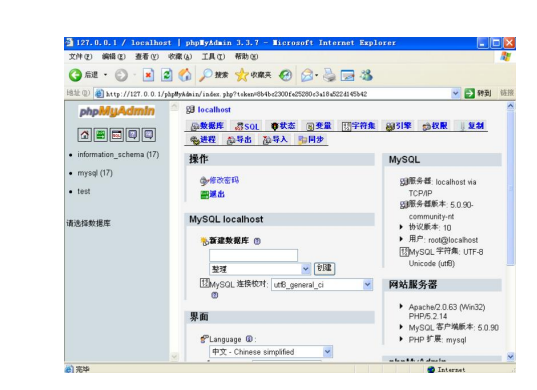
1. Phpnow安装

在虚拟机里安装phonow

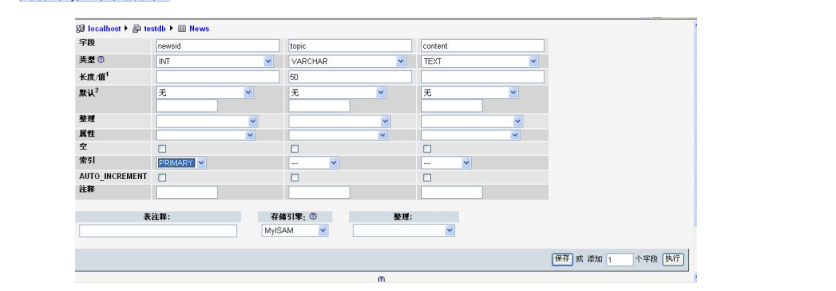


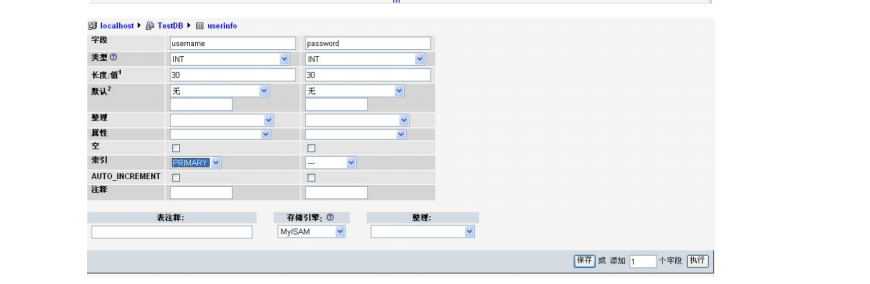


1. 打开默认界面，进入数据库管理系统

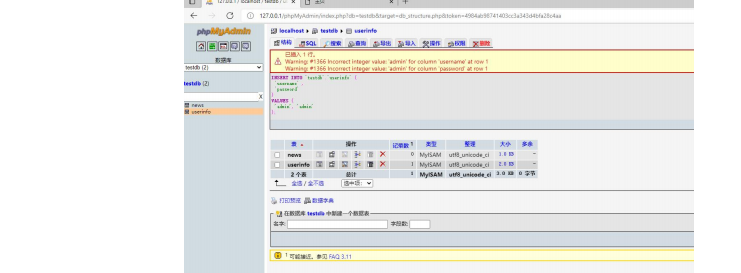


新建 TestDB 数据库，以及 News（newsid, topic, content）、userinfo（username, password） 两张表：





创建成功后向表中插入数据



1. 编写HTML文件

新建一个 html 文件



编写文件：

**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>

这个HTML文件定义了一个登录表单，使用POST方法将用户提交的数据发送到服务器端的loginok.php文件进行处理。表单的ID和名称均为form1，内部采用表格布局（宽度900px，无边框），包含用户名输入框、密码输入框和提交按钮。

用户名输入框的类型为text，名称和ID均为username，左侧单元格显示标签文字“username”。密码输入框的类型为password，名称和ID均为pwd，左侧显示“password”标签，输入内容会以掩码形式隐藏。所有输入控件都包裹在<label>标签内，提高了可访问性，但未设置required属性，因此前端不会强制要求用户填写。

提交按钮位于第三行的右侧单元格，类型为submit，显示文字为“提交”，名称设为Submit。当用户填写信息并点击提交后，浏览器会将数据（username=用户输入&pwd=密码）以POST请求发送至loginok.php，由服务器端进行验证和处理。

**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

}

?>

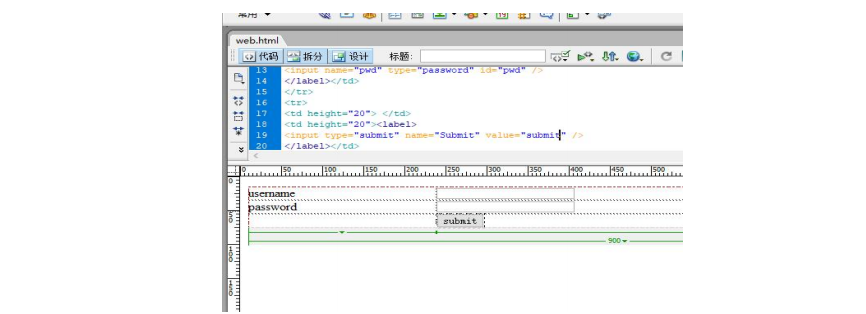
这段 PHP 代码是一个用户登录验证脚本，用于检查用户提交的用户名和密码是否与数据库中的记录匹配。

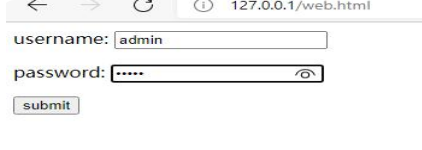
首先，代码初始化了一个登录状态标志 $loginok=0，然后使用 mysql\_connect() 函数建立与 MySQL 数据库的连接，连接参数为本地主机（localhost）、用户名 root 和密码 123456。接着，代码通过 $\_POST 超全局数组获取用户提交的用户名和密码，分别存储在 $username 和 $pwd 变量中。

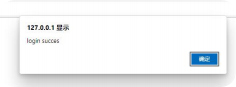
代码执行查询时使用了 mysql\_db\_query() 函数（该函数在 PHP 5.5.0 中已被废弃），在 testDB 数据库中查询匹配的用户记录。如果查询结果通过 mysql\_fetch\_array() 获取到数据行，则将 $loginok 标志设为 1 表示登录成功。之后代码释放了查询结果并关闭了数据库连接。

根据登录状态的不同，代码会输出不同的 JavaScript 响应：登录成功时弹出"login success"提示并跳转到 sys.php 页面；登录失败时弹出"login failed"提示并返回上一页。

尝试运行：







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);

?>

这段代码是一个新闻管理系统的后台页面（sys.php），主要实现了新闻的添加和展示功能，同时提供了删除新闻的入口。页面采用HTML表格布局，顶部是一个新闻添加表单，下方是新闻列表展示区域。在PHP部分，代码首先建立了与MySQL数据库的连接，使用root账户和123456密码连接本地数据库服务器。

页面顶部的表单使用POST方法提交到add.php，包含两个输入字段：新闻标题（单行文本框）和新闻内容（多行文本域）。表单下方通过水平线分隔，接着是新闻列表展示区域。新闻列表采用表格形式显示，包含三列：新闻序号、新闻标题和删除操作。PHP代码通过执行"select \* from news"查询从testDB数据库获取所有新闻记录，并使用while循环逐行显示。每条新闻的删除操作通过超链接指向del.php，并在URL中附带新闻ID作为参数（newsid=xxx）。

**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

}

?>

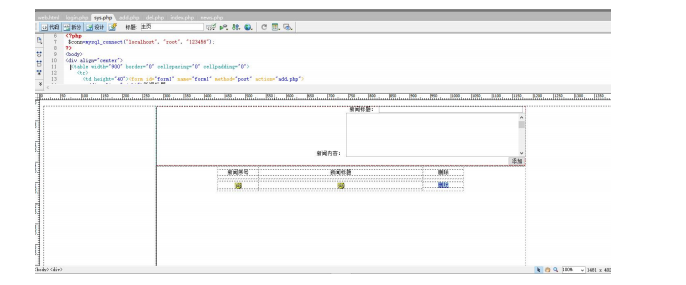
这段 PHP 代码 (add.php) 是一个新闻添加功能的处理脚本，负责将用户在 sys.php 页面表单中提交的新闻数据插入到数据库中。

代码首先建立与 MySQL 数据库的连接，使用 mysql\_connect() 函数连接到本地主机的 MySQL 服务器，用户名是 root，密码是 123456。然后通过 mysql\_select\_db() 选择名为 testDB 的数据库。

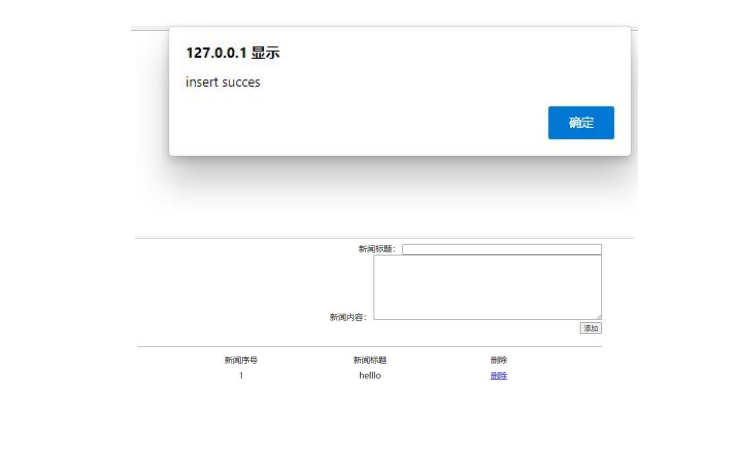
接下来，代码通过 $\_POST 超全局数组获取用户提交的表单数据，$topic 变量接收新闻标题，$content 变量接收新闻内容。这些用户输入被直接拼接到 SQL 插入语句中，构成了 $SQLStr 变量

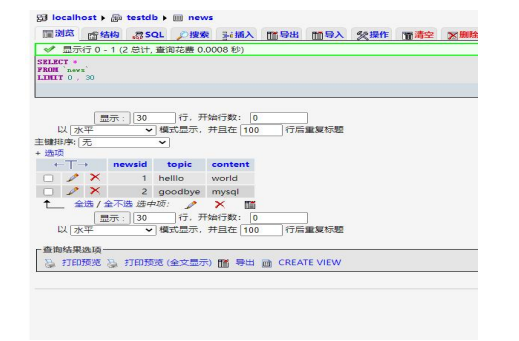
代码执行插入操作后，会根据操作结果显示不同的 JavaScript 提示：如果插入成功 ($result 为真)，则弹出"insert success"提示并重定向回 sys.php 页面；如果失败，则弹出"insert failed"提示并返回上一页。最后，无论操作成功与否，都会关闭数据库连接。

运行：









**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

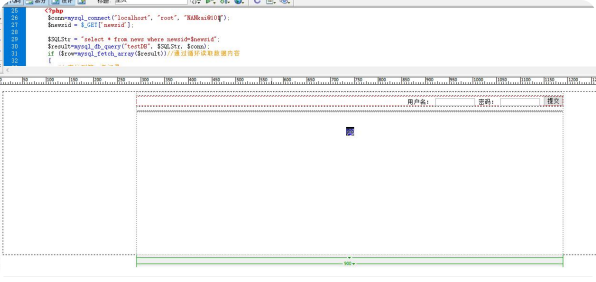
}

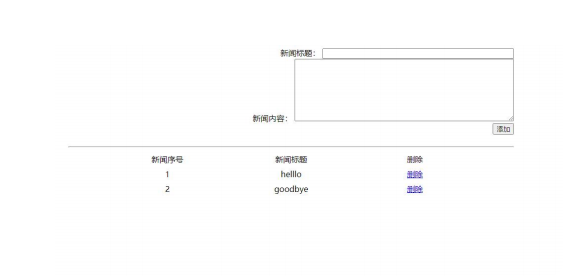
?>

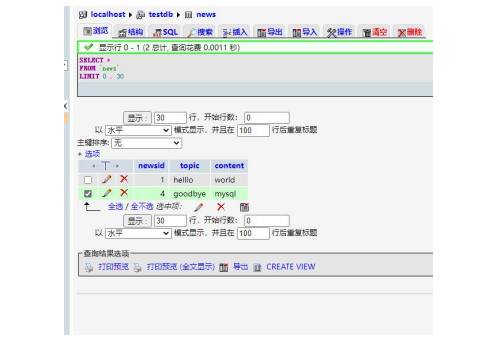
news.php 是一个新闻详情展示页面，主要功能是显示单条新闻的完整内容。页面顶部包含一个登录表单（用户名和密码输入框），下方通过 newsid 参数从数据库查询并显示特定新闻的标题和内容。用户访问时需要传入 newsid 参数（如 news.php?newsid=1）来指定要查看的新闻。

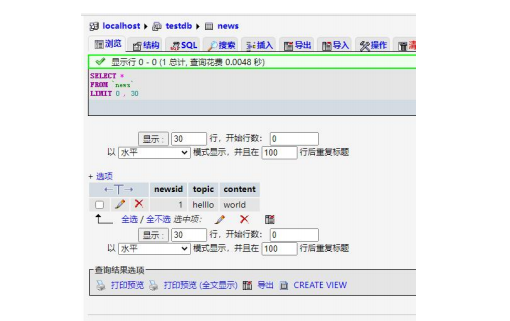
del.php 是新闻删除功能的处理脚本，它接收 newsid 参数，从数据库中删除对应的新闻记录。删除成功后显示"delete success"提示并返回新闻管理页面（sys.php），失败则提示"delete failed"并返回上一页。

运行代码：









再次运行代码：

**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

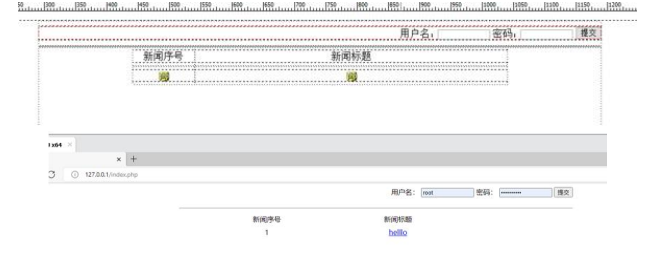
mysql\_free\_result($result);

// 关闭连接

mysql\_close($conn);

?>

最终结果：



**心得体会：**

通过这次 实验，我对 WEB 开发全流程有了更清晰的认知，从搭建 PHPnow 集成环境开始，到在 phpMyAdmin 中设计数据库表结构，再到用 Dreamweaver 编写 HTML 页面和 PHP 脚本，整个过程让我直观感受到前端页面、后端逻辑与数据库之间的协同运作。比如在编写登录表单时，我理解了 HTML 如何通过 POST 方法将用户数据传递给 PHP 处理，而 PHP 又如何通过数据库连接函数与 MySQL 交互的前后端的联动机制。

在数据库操作的实践中，我深入掌握了用 PHP 实现增删查改的核心逻辑。无论是 loginok.php 中通过 SQL 查询验证用户身份，还是 sys.php 里用循环遍历新闻列表，亦或是 add.php 和 del.php 完成数据的插入与删除，每一个步骤都让我对 SQL 语句与 PHP 变量的结合有了更熟练的运用。