

《WEB 开发实践》实验报告

姓名：汤清云 学号：2013536 班级： 1075

实验名称：

WEB 开发实践

实验要求：

利用 php 编写简单的数据库插入、删除、查询操作的示例。

进一步了解 WEB 开发实践的细节。

实验知识点：

URL 即标识以 http 或 https 协议请求的资源。

HTML 语言 head 应包括以下元素：

标签	描述
<head>	定义了文档的信息
<title>	定义了文档的标题
<base>	定义了页面链接标签的默认链接地址
<link>	定义了一个文档和外部资源之间的关系
<meta>	定义了 HTML 文档中的元数据
<script>	定义了客户端的脚本文件
<style>	定义了 HTML 文档的样式文件

其语言主体应被包括在<body> </body>中。

http 属于无状态通信协议，服务器不会记录请求发自哪个浏览器。故而服务器引入 session 机制，使用类似散列表的方式来记录信息；而客户端引入 cookie 机制记录这些请求是否发送给同一个服务器。一般而言，服务器使用 session 维护客户端的会话状态，客户端使用 cookie 存储当前客户端的会话状态的 session id，在使用时将它传递给服务器，让服务器找到对应状态并呈现给客户端。

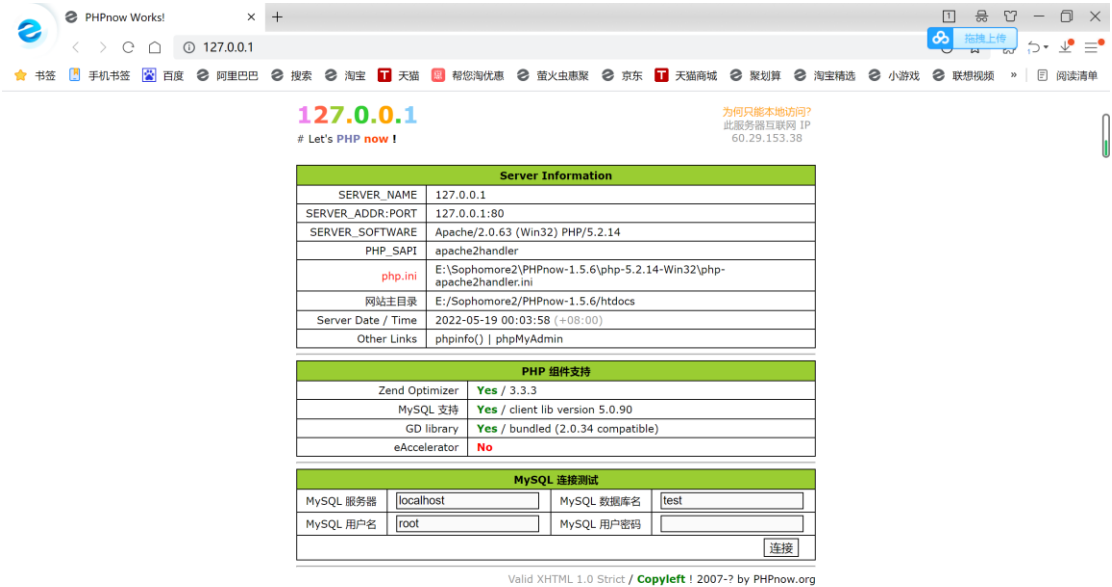
PHP 语言是一种免费的脚本语言，所有 PHP 语句都被包括在 PHP 标记里，PHP 标记外的语句都会直接输出。PHP 的四种标记如下：

四种标记：		
标记	解释	示例
<?php ?>	标准 php 标记	<?php echo \$variable; ?>
<script language="php"> </script>	长标记	<script language="php"> echo \$variable; </script>
<? ?>	短标记	<? echo \$variable; ?>

<% %>	仿 ASP	<%= \$variable; %>
-------	-------	--------------------

实验过程:

1. 安装 PHPnow: 使用 cmd.exe 以管理员身份运行 Init.exe, 完成各项任务安装。此时弹出以下界面, 访问资源为 hpdocs-index.php 的默认资源



2. 输入密码 123456 查看数据库连接是否正常:

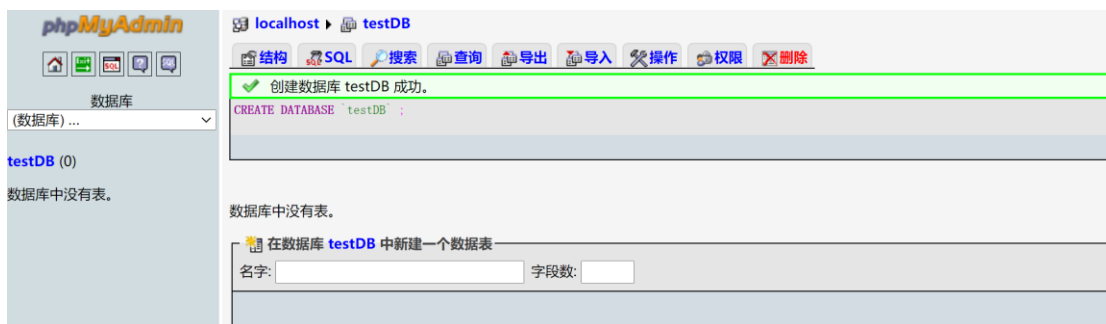
Other Links | phpinfo() | phpMyAdmin

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

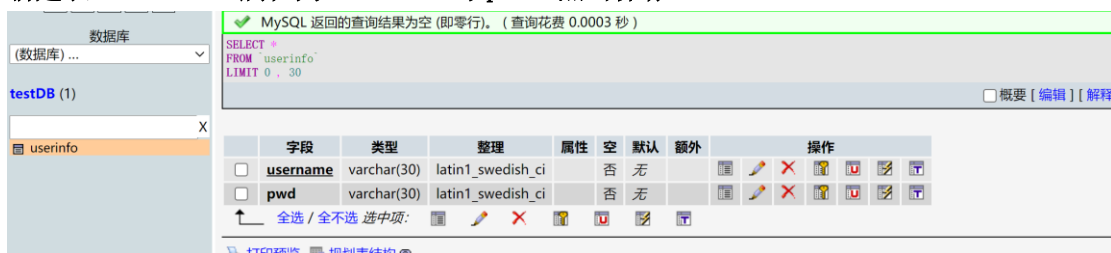
MySQL 连接测试			
MySQL 服务器	<input type="text" value="localhost"/>	MySQL 数据库名	<input type="text" value="test"/>
MySQL 用户名	<input type="text" value="root"/>	MySQL 用户密码	<input type="text"/>
			<input type="button" value="连接"/>

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

3. 点击 phpMyAdmin, 用户名密码为 root/123456, 登录进数据库管理系统。创建一个新的数据库:



4. 新建表 userinfo, 属性为 username 与 pwd, 点击保存:



5. 插入数据, admin/admin



6. 使用 DW 软件创建 html 文件, 保存在 hpdocs 文件夹下。修改 html 文件文档。添加表单 form1, 添加 action, 添加表单对象为文本。添加 username, password 文本以及提交按钮。具体代码如下:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>test1</title>
</head>

<body>

<form action="loginok.php" method="get" name="form1" id="form1">
  <p>
    <label for="password">username:&nbsp;</label>
    <input type="text" name="username" id="username">
  </p>
```

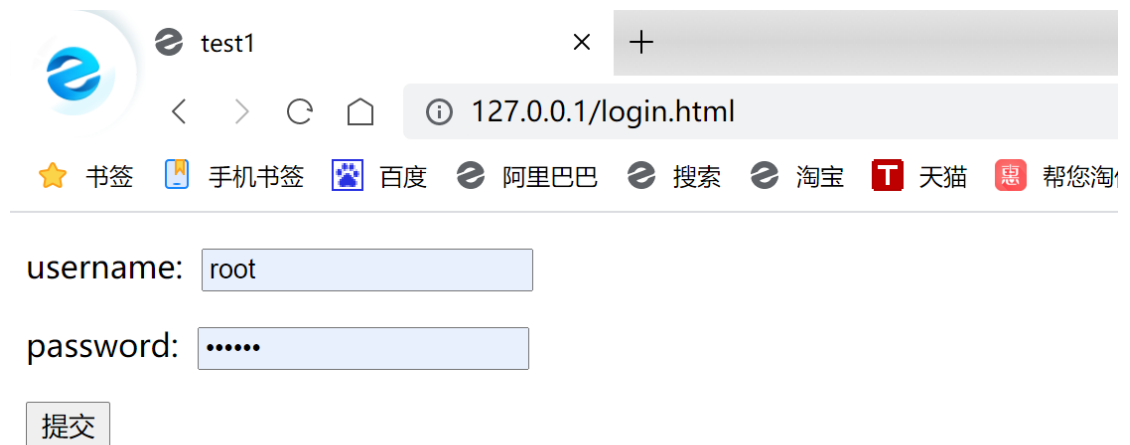
```

<p>
<label for="password">password:&nbsp;</label>
<input type="password" name="password" id="password">
</p>
<p>
<input type="submit" name="submit" id="submit" value="提交">
</p>

</form>
</body>
</html>

```

实现效果为:



7. 设置 username 为 testget, 密码为 testget 得出如下效果: 显然 get 模式会将具体的值显示在 URL 中。



Not Found

8. 在htdocs文件夹下新建 login.php 文件, 打开并编辑代码如下:

```

<?php
$username=$_GET['username'];变量 username 获取标签为 username 的内容, 由于 html 文件中方式为 get, 故此处可以得到。
$password=$_GET['password'];
echo $username;输出变量 username 的值
?>

```

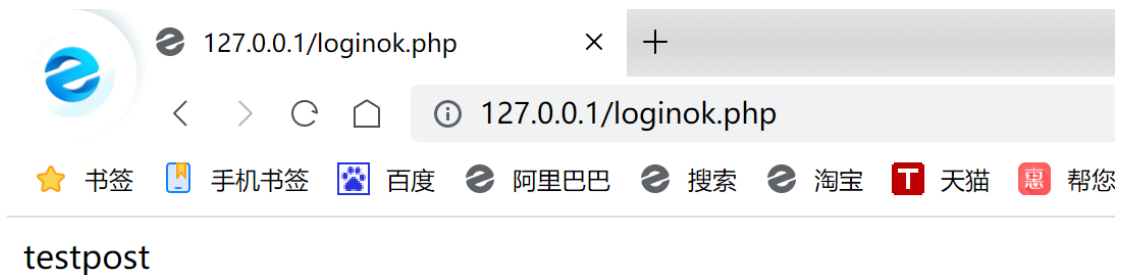
9. 再次刷新网页即可得到如下结果, 输出了我们定义的 username testget



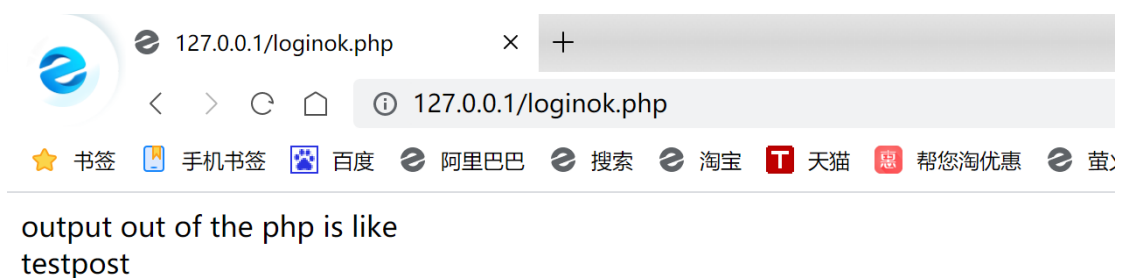
10. 修改 html 文件中方式为 post，再次回到登陆页面并输入用户名密码为 testpost/testpost，结果如下：



11. 再次将 php 中获取方式改为 POST 方式得结果：



12. 在 php? 范围外的文本将被直接输出：



13. 添加构造 SQL 语句如下：

```
<?php

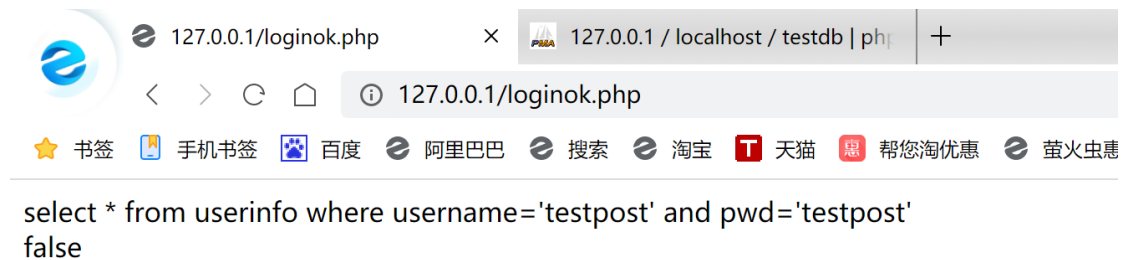
$conn=mysql_connect("localhost","root","123456");//连接数据库
$username=$_POST['username'];
$password=$_POST['password'];
$SQLstr="select * from userinfo where username='$username' and
pwd='$password' ";
echo $SQLstr;
```

```

$result=mysql_db_query("testDB",$SQLstr,$conn);//执行数据库语句
//获取查询结果
if($row=mysql_fetch_array($result))
    echo "<br>OK<br>";
else
    echo "<br>false<br>";
//释放资源
mysql_free_result($result);
//关闭连接
mysql_close($conn);
?>

```

结果如下:



若输入之前插入的数据 admin/admin, 结果为:



14. 修改代码实现以下功能: 输入正确的用户名密码跳转至 sys.html 界面, 否则回到历史界面

代码为:

```

<?php
$isok=0;
$conn=mysql_connect("localhost","root","123456");//连接数据库
$username=$_POST['username'];
$password=$_POST['password'];
$SQLstr="select * from userinfo where username='$username' and
pwd='$password' ";
echo $SQLstr;
$result=mysql_db_query("testDB",$SQLstr,$conn);//执行数据库语句
//获取查询结果

```

```

if($row=mysql_fetch_array($result))
    $isok=1;

//释放资源
mysql_free_result($result);
//关闭连接
mysql_close($conn);
if($isok==1)
{
    ?>
    <script language="javascript">
        alert("ok!");//弹出对话框
        window.location.href="sys.php";//跳转至该页面

    </script>

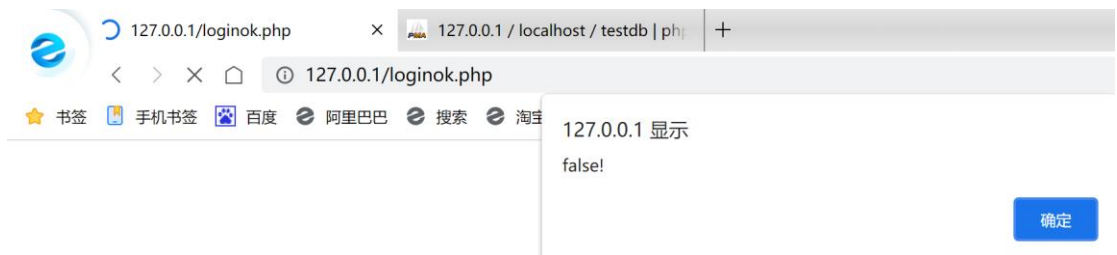
    <?//使用问号打断和重新开始，使得两端之间的代码不解释运行
}
else
{
    ?>
    <script language="javascript">
        alert("false!");
        history.back();//回去历史界面

    </script>
    <?//使用问号打断和重新开始
}
?>

```

结果如下：





15. 在 testdb 数据库中新建 news 表, 属性为 newsid, topic, content。新建 sys.php 页面, 布局为 topic 为标签, 一个文本输入框, content 为标签, 一个文本区域用于输入。具体代码如下:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>test1</title>
</head>

<body>

<form action="addok.php" method="post" name="form1" id="form1">
  <p>
    <label for="password">topic:&nbsp;</label>
    <input type="text" name="topic" id="topic">
  </p>
  <p>
    <label for="password">content:&nbsp;</label>
    <br>
    <label for="content">&nbsp;</label>
    <textarea name="content" cols="60" rows="8"
id="content"></textarea>
  </p>
  <p>
    <input type="submit" name="submit" id="submit" value="提交">
  </p>

</form>
</body>
</html>
```

新建 addok.php 文件, 实现功能为对 news 表实现插入功能, 其具体功能为读取 sys.php 页面中获取的 topic 和 content 值, 并且插入到 news 表中。实现代码如下:

```
<?php
$conn=mysql_connect("localhost","root","123456");//连接数据库
```



```

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);//执行数据库语句

//关闭连接
mysql_close($conn);
if($result)//执行操作是否成功
{
    ?>
    <script language="javascript">
        alert("add ok!");//弹出对话框
        window.location.href="sys.php";//跳转至该页面

    </script>

    <?//使用问号打断和重新开始，使得两端之间的代码不解释运行
}
else
{
    ?>
    <script language="javascript">
        alert("add false!");
        history.back();//回去历史界面

    </script>
    <?//使用问号打断和重新开始
}
?>

```

在页面中成功登陆后即可实现插入操作：



The screenshot shows a web application interface for a database named 'testdb (2)'. On the left, there is a sidebar with a search bar and a list of tables: 'news' and 'userinfo'. The main area displays the 'news' table. At the top, there are controls for displaying rows: '显示: 30 行, 开始行数: 0'. Below this, there is a dropdown menu set to '水平' (Horizontal) and a checkbox for '模式显示, 并且在 100 行后重复标题'. The table has three columns: 'newsid', 'topic', and 'content'. The first row shows 'newsid' as 1, 'topic' as 'test insert', and 'content' as garbled text: 'æµ·è·æ~á|æ`ÔáŠŸá@žçZ°æ□' ä...¥áŠŸëf½â€'. At the bottom, there are navigation icons and a status bar.

（此处 content 乱码是因为输入为中文。）

16. 新建 html 文件，另存为为 news.php，实现功能为显示出 news 表中所存信息的 id 和 topic，在 topic 处显示链接。具体实现代码如下：

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>无标题文档</title>
</head>
<body>

<table width="600" border="1" align="center">
  <tr>
    <td><p>id</p></td>
    <td>topic</td>
  </tr>
  <?php
    $conn=mysql_connect("localhost","root","123456");//连接数据库
    $SQLStr="select newsid,topic 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
      while($row=mysql_fetch_row($result))
      {

        ?>
        <tr>
          <td><?php echo $row[0];?></td>

          <td><a href="see.php?newsid=<?php echo $row[0];?>">
            <?php echo $row[1];?>
          </a></td>
        </tr>
        <?php

          }

        }
        ?>
      </table>
```

```
</body>
</html>
```



17. 新建 see.php，实现对该信息的查看。

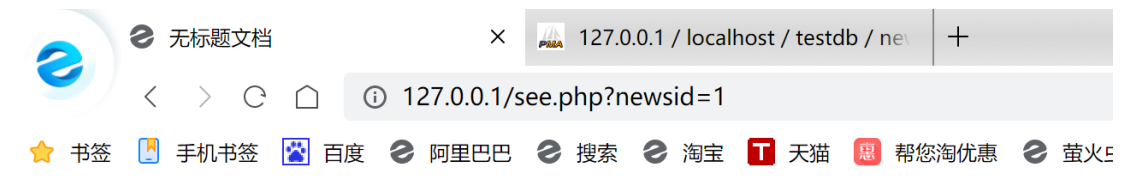
代码如下：

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>无标题文档</title>
</head>
<body>

<?php
$conn=mysql_connect("localhost","root","123456");//连接数据库
$id=$_GET['newsid'];
$SQLStr="select topic,content from news where newsid= '$id'";
$result=mysql_db_query("testDB",$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
    while($row=mysql_fetch_row($result))
    {
        ?>
        topic:<?php echo $row[0];?><br>
        content:<?php echo $row[1];?><br>
    }
}
?>

</body>
```

</html>



18. 再次插入一条语句, topic 为 test delete, content 为测试是否成功实现删除功能。



id	topic
1	test insert
2	delete

修改 addok.php 中 sql 语句为删除, 代码变为:

```
<?php
$conn=mysql_connect("localhost","root","123456");//连接数据库
mysql_select_db("testDB");
$topic=$_POST['topic'];
$content=$_POST['content'];
$SQLstr="delete from news where news.topic='$topic'";
echo $SQLstr;
$result=mysql_query($SQLstr);//执行数据库语句

//关闭连接
mysql_close($conn);
if($result)//执行操作是否成功
{
    ?>
    <script language="javascript">
        alert("delete ok!");//弹出对话框
        window.location.href="sys.php";//跳转至该页面

    </script>
```

```

        <?//使用问号打断和重新开始，使得两端之间的代码不解释运行
    }
    else
    {
        ?>
        <script language="javascript">
            alert("add false!");
            history.back();//回去历史界面

        </script>
        <?//使用问号打断和重新开始
    }
?>

```

在 addok 界面的 topic 输入 delete，结果如下：



可以看出语句被删除，删除语句测试成功。

心得体会：

学习了网站的初步构造，了解了 php 语言的编写和网页的布局。