lab11 报告

姜尔玲 17307130291

概述

lab11的任务是学习使用cookie和session，维护用户的登录状态。

具体过程

登录验证

提交表单后，登录验证时调用的函数是validlogin()，在这个函数中，我们使用PDO对象进行数据库的访问，通过对得到的结果的rowCount()进行判断，对用户填入的信息进行验证。

使用cookie

在登录验证成功之后，使用如下setcookie()函数设置cookie：

```
$expiryTime = time()+60*60*24;
setcookie("UserName", $_POST['username'], $expiryTime);
```

在log out的时候，使用如下方法删除该cookie：

```
setcookie("Username", "", -1);
header("Location: ".$_SERVER['HTTP_REFERER']);
```

需要注意的一个点在于：cookie是默认需要刷新才会真正发挥作用的，即具体的`$_COOKIE['Username']`才不是空值。所以在登录的代码中，我增加了一段手动刷新页面的代码如下，保证登录之后、再次刷新之前页面可以正常显示已登录的界面：

```
$url = get_current_url();
header("location:".$url);
```

其中get_current_url()函数代码如下：

```php
function get_current_url(){
    $current_url='http://';
    if(isset($_SERVER['HTTPS'])&&$_SERVER['HTTPS']=='on'){
        $current_url='https://';
    }
    if($_SERVER['SERVER_PORT']!='80'){

 $current_url.=$_SERVER['SERVER_NAME'].':'.$_SERVER['SERVER_PORT'].$_SERVER['R
EQUEST_URI'];
    }else{
        $current_url.=$_SERVER['SERVER_NAME'].$_SERVER['REQUEST_URI'];
    }
    return $current_url;
}
```

- 优点：
  - 易于使用和实现。
  - 存储在用户的计算机上，不需要服务器资源。
  - 持久性；在客户端可以持续的时间长，且服务器的崩溃不会对其产生影响。
  - 透明性；用户并不知道cookie的工作方式。
  - 响应速度快。
  - 易于管理；大多数浏览器都可以让用户轻松清楚浏览历史记录，只需需转到工具，清除历史记录并选择Cookie即可。Cookie存储在用户硬盘驱动器上的cookie.txt下的文本文件中，因为它是一个文本文件，我们可以使用任何查看器或文本编辑器来显示，编辑和删除它们。
- 缺点：
  - 隐私问题；web浏览器会跟踪所访问过的所有网站，并且很有可能，第三方可以访问这些存储的信息并且提供广告等服务。
  - 不安全；cookie是以明文形式存储的，任何人都可以打开并篡改。
  - 难以解密；cookie的大小有限制，智能存储简单字符串信息。
  - 可以被禁用；用户可以在浏览器中选择禁用，这有可能会带来一些问题。

使用session

在这里，session可以看作是被包装的cookie，均是对用户状态进行存储。

在php文件的开头，需要写一行`session_start();`创建并在之后session。同cookie，在登录验证成功之后，我们使用如下代码创建具体的session：

```php
$_SESSION['Username']=$_POST['username'];
```

在log out时，使用如下代码删除已有的session信息：

```php
unset($_SESSION['Username']);
```

- 优点：

    - 安全性较高；session的数据是存储在服务器端，客户端存储的只是数据的地址。
- 缺点：
    - 占用服务器资源；数据存储在服务器端。
    - 禁用cookie之后，session同样会失效。