CAN302 – Technologies for e-commerce

Division of communication and network
Xi’an Jiaotong-Liverpool University

Lab 8 – Login and session

Aim

The aim of this lab is to learn the login function and session.

Tips:

1. If you are not sure why you are doing something, ask a TA. This is what they are here for.

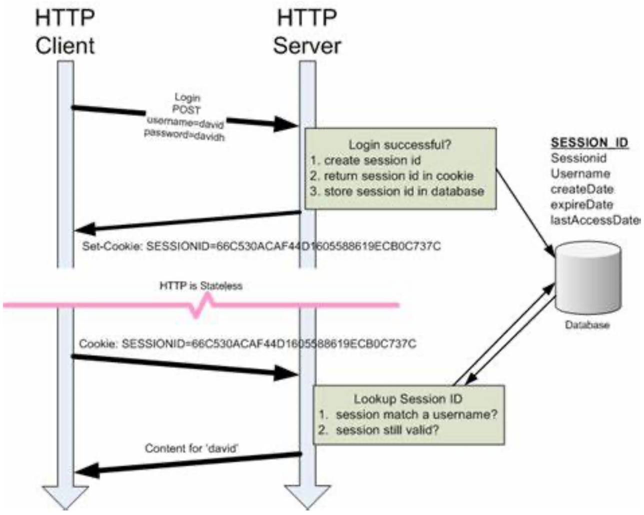
2. The M-Dev-Store online videos are good references while our labs have different focus. If you want to be an expert, you are recommended take both labs and on-line videos.

3. The forums @ LMO are available for questions and discussions.

4. These labs are expected take more than the 2 allocated hours. You should complete them in your own time before the next lab. Practice makes perfect!

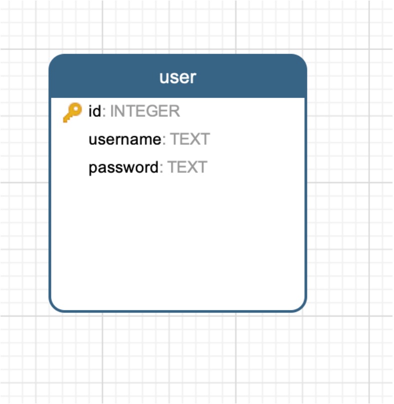
Cookie and Session:

1. Due to the stateless of HTTP, cookie or session are needed for state related scenario. In last lab, we use cookie to identify if the visiting is by the same user (the same browser on the same device) or not. Since people can easily get the information in the cookie and even modify it. It is too risky to store the important data like user id in cookie.
   Session is a global variable stored on the server. Each session is assigned a unique id which is used to retrieve stored values and the session id is stored as a cookie on the user’s computer. Even the user can see the session id, but no sensitive data can be tamped. Even the session id was leakage, the server still can handle the risk by a proper design. For example, the ip address of client computer could be stored in the session. When the seesion id from a different ip address, the session will be destroyed.
   For the login function, session rather than only cookie should be used. A typical interaction is as:



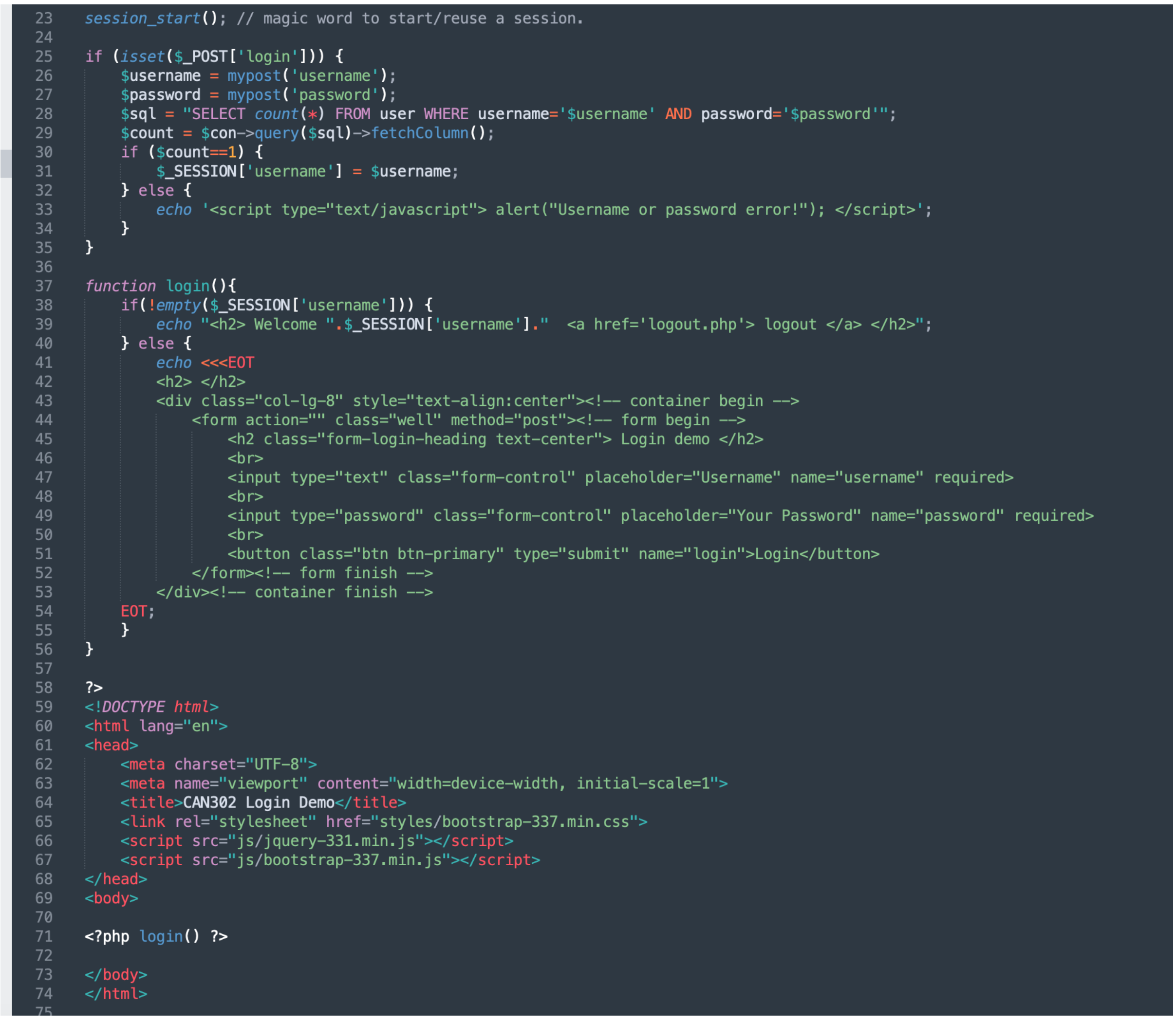
Login sample:

2. We need a table to record the username and password. We can create a simple table only for demo purpose as:



user
id: INTEGER
username: TEXT
password: TEXT

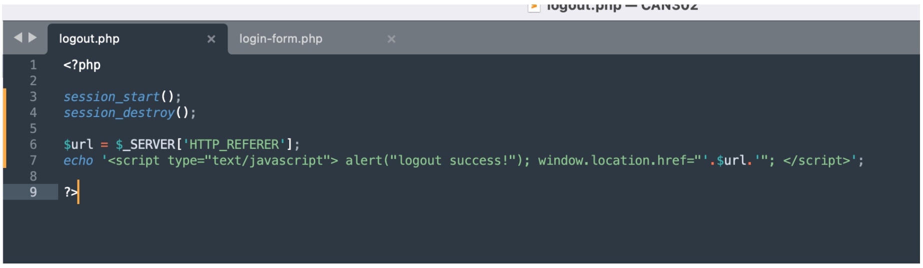
A login demo is as following:



```
session_start(); // magic word to start/reuse a session.

if (isset($_POST['login'])) {
    $username = mypost('username');
    $password = mypost('password');
    $sql = "SELECT count(*) FROM user WHERE username='$username' AND password='$password'";
    $count = $con->query($sql)->fetchColumn();
    if ($count==1) {
        $_SESSION['username'] = $username;
    } else {
        echo '<script type="text/javascript"> alert("Username or password error!"); </script>';
    }
}

function login(){
    if(!empty($_SESSION['username'])) {
        echo "<h2> Welcome ".$_SESSION['username']."  <a href='logout.php'> logout </a> </h2>";
    } else {
        echo <<<EOT
        <h2> </h2>
        <div class="col-lg-8" style="text-align:center"><!-- container begin -->
            <form action="" class="well" method="post"><!-- form begin -->
                <h2 class="form-login-heading text-center"> Login demo </h2>
                <br>
                <input type="text" class="form-control" placeholder="Username" name="username" required>
                <br>
                <input type="password" class="form-control" placeholder="Your Password" name="password" required>
                <br>
                <button class="btn btn-primary" type="submit" name="login">Login</button>
            </form><!-- form finish -->
        </div><!-- container finish -->
    EOT;
    }
}

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>CAN302 Login Demo</title>
    <link rel="stylesheet" href="styles/bootstrap-337.min.css">
    <script src="js/jquery-331.min.js"></script>
    <script src="js/bootstrap-337.min.js"></script>
</head>
<body>

<?php login() ?>

</body>
</html>
```

The function to open database is the same. "session_start" is the PHP function to start or re-use the session. The rest of the page can use the session. "$_SESSION" is a global array. Similar to the form function in lab2, this page will check the login data. If there is login data, it will try to compare the username and password stored in the database. If the record number is 1, "$_SESSION["username"]" will be set to current user's name. In the "login" function, we can determine the login status by checking the value of "$_SESSION["username"]". If the user is login, show a welcome, otherwise a login form would be shown. More info can be stored in "$_SESSION" array if they are related to a session.

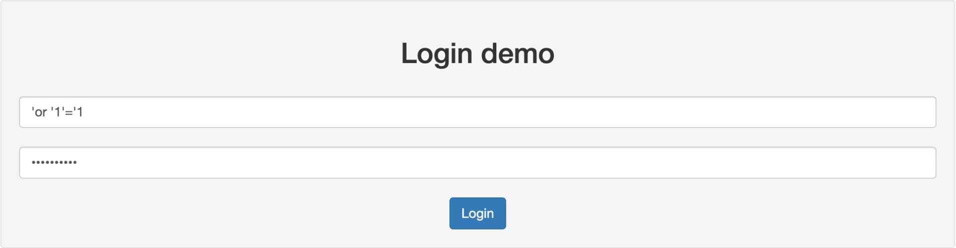
If we want to logout the user, the code is as simple as:



```
<?php

session_start();
session_destroy();

$url = $_SERVER['HTTP_REFERER'];
echo '<script type="text/javascript"> alert("logout success!"); window.location.href="'.$url.'"; </script>';

?>
```

The function "session_destroy" would destroy the current session. The rest JS code would re-redirect to the referer page.

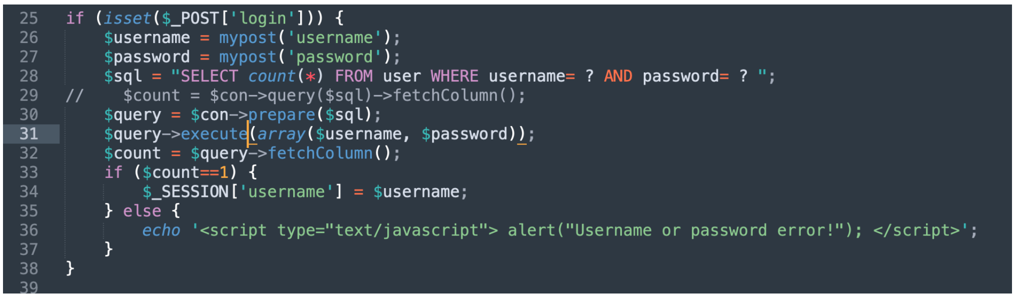
Does it safe enough? No at all! Let's try a SQL injection. Set the username and password or to be 'or '1'='1 as following:



'or '1'='1

••••••••••

You can pass the login easily. A way to against this attack is as:

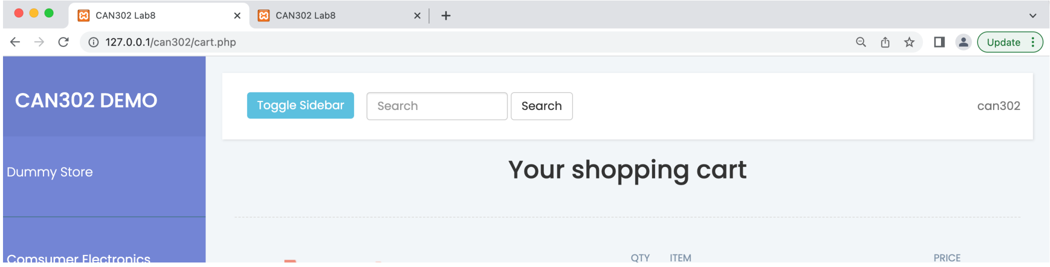


```
if (isset($_POST['login'])) {
    $username = mypost('username');
    $password = mypost('password');
    $sql = "SELECT count(*) FROM user WHERE username= ? AND password= ? ";
//    $count = $con->query($sql)->fetchColumn();
    $query = $con->prepare($sql);
    $query->execute(array($username, $password));
    $count = $query->fetchColumn();
    if ($count==1) {
        $_SESSION['username'] = $username;
    } else {
        echo '<script type="text/javascript"> alert("Username or password error!"); </script>';
    }
}
```

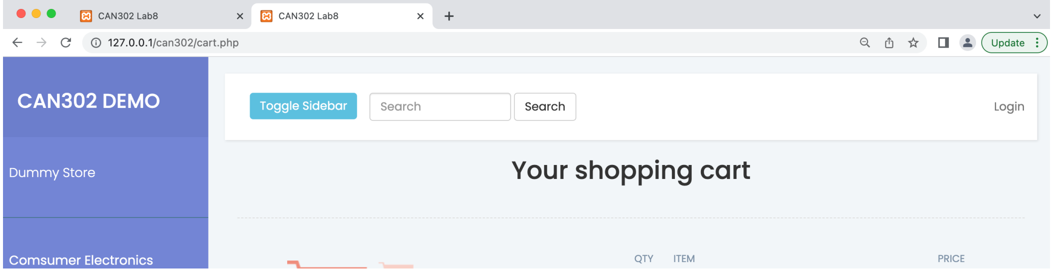
BE CAREFUL:
It is only a demo to show the SQL injection. The above code may have other weaknesses and should **NOT** be used in any real project.

Integrate the login to demo store:

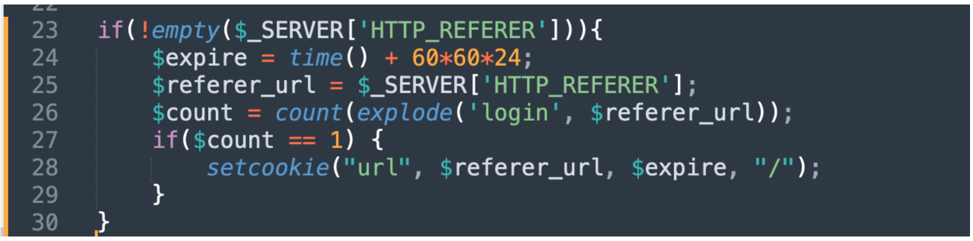
3. In the shopping cart page, the session value should be tested to check if the user login or not. If the user login, the username can be shown as:



Otherwise, the user can click the login to "login-form.php".



How to redirect the user back to the original page? It can be done by cookie. Some tips as:

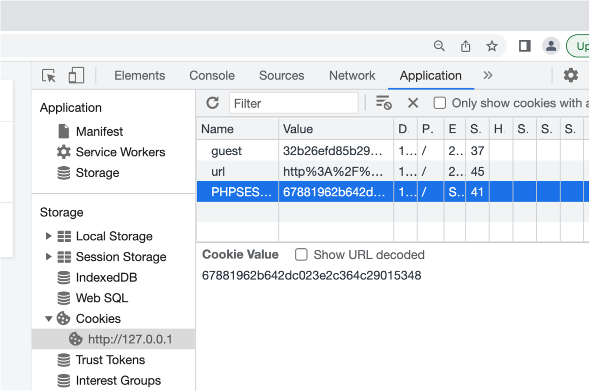


```php
if(!empty($_SERVER['HTTP_REFERER'])){
    $expire = time() + 60*60*24;
    $referer_url = $_SERVER['HTTP_REFERER'];
    $count = count(explode('login', $referer_url));
    if($count == 1) {
        setcookie("url", $referer_url, $expire, "/");
    }
}
```

Home work

4. Try to integrate the logout function also.

5. Please check the value of session id. Whether the user login or not, it won't change. Try to figure out why and find a way to change it.



Name	Value	D.	P.	E	S.	H.	S.	S.	S.
guest	32b26efd85b29…	1…	/	2…	37				
url	http%3A%2F%…	1…	/	2…	45				
PHPSES…	67881962b642d…	1…	/	S…	41				

Cookie Value □ Show URL decoded

67881962b642dc023e2c364c29015348

Created Fall 2022-2023
By Dr Ping Zhang