**Batch: B-1 Roll No.: 16010122104**

**Experiment No. 06**

|  |
| --- |
| **TITLE:** Cookies and Session in PHP |

**AIM: The aim of this experiment is to learn how to create and destroy cookies and sessions in PHP**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

The expected outcome of the experiment is understanding of how to effectively manage cookies and sessions in PHP.

Books/ Journals/ Websites referred:

1. .

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Problem Statement: Description of the application implemented with output**:

1. Create a secure login page with fields for username and password. Implement authentication of user credentials. Upon successful login, establish a session and set cookies, then redirect the user to a welcome page.

2. Upon refreshing the welcome page, ensure the session and cookies are properly destroyed.

Implementation and screen shots of output:

<?php

session\_start();

error\_reporting(0);

include('includes/config.php');

if($\_SESSION['alogin']!=''){

$\_SESSION['alogin']='';

}

if(isset($\_POST['login']))

{

$uname=$\_POST['username'];

$password=md5($\_POST['password']);

$sql ="SELECT UserName,Password FROM admin WHERE UserName=:uname and Password=:password";

$query= $dbh -> prepare($sql);

$query-> bindParam(':uname', $uname, PDO::PARAM\_STR);

$query-> bindParam(':password', $password, PDO::PARAM\_STR);

$query-> execute();

$results=$query->fetchAll(PDO::FETCH\_OBJ);

if($query->rowCount() > 0)

{

$\_SESSION['alogin']=$\_POST['username'];

echo "<script type='text/javascript'> document.location = 'dashboard.php'; </script>";

} else{

    echo "<script>alert('Invalid Details');</script>";

}

}

?>

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="utf-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1">

        <title>Admin Login</title>

        <link rel="stylesheet" href="css/bootstrap.min.css" media="screen" >

        <link rel="stylesheet" href="css/font-awesome.min.css" media="screen" >

        <link rel="stylesheet" href="css/animate-css/animate.min.css" media="screen" >

        <link rel="stylesheet" href="css/prism/prism.css" media="screen" > <!-- USED FOR DEMO HELP - YOU CAN REMOVE IT -->

        <link rel="stylesheet" href="css/main.css" media="screen" >

        <script src="js/modernizr/modernizr.min.js"></script>

    </head>

    <body class="">

        <div class="main-wrapper">

            <div class="">

                <div class="row">

 <h1 align="center">Student Result Management System</h1>

                         <div class="col-lg-3"></div>

                    <div class="col-lg-6">

                        <section class="section">

                            <div class="row mt-40">

                                <div class="col-md-10 col-md-offset-1 pt-50">

                                    <div class="row mt-30 ">

                                        <div class="col-md-11">

                                            <div class="panel">

                                                <div class="panel-heading">

                                                    <div class="panel-title text-center">

                                                        <h4>Admin Login</h4>

                                                    </div>

                                                </div>

                                                <div class="panel-body p-20">

                                                    <form class="form-horizontal" method="post">

                                                        <div class="form-group">

                                                            <label for="inputEmail3" class="col-sm-2 control-label">Username</label>

                                                            <div class="col-sm-10">

                                                                <input type="text" name="username" class="form-control" id="inputEmail3" placeholder="UserName">

                                                            </div>

                                                        </div>

                                                        <div class="form-group">

                                                            <label for="inputPassword3" class="col-sm-2 control-label">Password</label>

                                                            <div class="col-sm-10">

                                                                <input type="password" name="password" class="form-control" id="inputPassword3" placeholder="Password">

                                                            </div>

                                                        </div>

                                                        <div class="form-group mt-20">

                                                            <div class="col-sm-offset-2 col-sm-10">

                                                                <button type="submit" name="login" class="btn btn-success btn-labeled pull-right">Sign in<span class="btn-label btn-label-right"><i class="fa fa-check"></i></span></button>

                                                            </div>

                                                        </div>

                                                    </form>

                                                </div>

                                            </div>

                                            <!-- /.panel -->

                                            <p class="text-muted text-center"><small>Student Result Management System</small></p>

                                        </div>

                                        <!-- /.col-md-11 -->

                                    </div>

                                    <!-- /.row -->

                                </div>

                                <!-- /.col-md-12 -->

                            </div>

                            <!-- /.row -->

                        </section>

                    </div>

                    <!-- /.col-md-6 -->

                </div>

                <!-- /.row -->

            </div>

            <!-- /. -->

        </div>

        <!-- /.main-wrapper -->

        <!-- ========== COMMON JS FILES ========== -->

        <script src="js/jquery/jquery-2.2.4.min.js"></script>

        <script src="js/jquery-ui/jquery-ui.min.js"></script>

        <script src="js/bootstrap/bootstrap.min.js"></script>

        <script src="js/pace/pace.min.js"></script>

        <script src="js/lobipanel/lobipanel.min.js"></script>

        <script src="js/iscroll/iscroll.js"></script>

        <!-- ========== PAGE JS FILES ========== -->

        <!-- ========== THEME JS ========== -->

        <script src="js/main.js"></script>

        <script>

            $(function(){

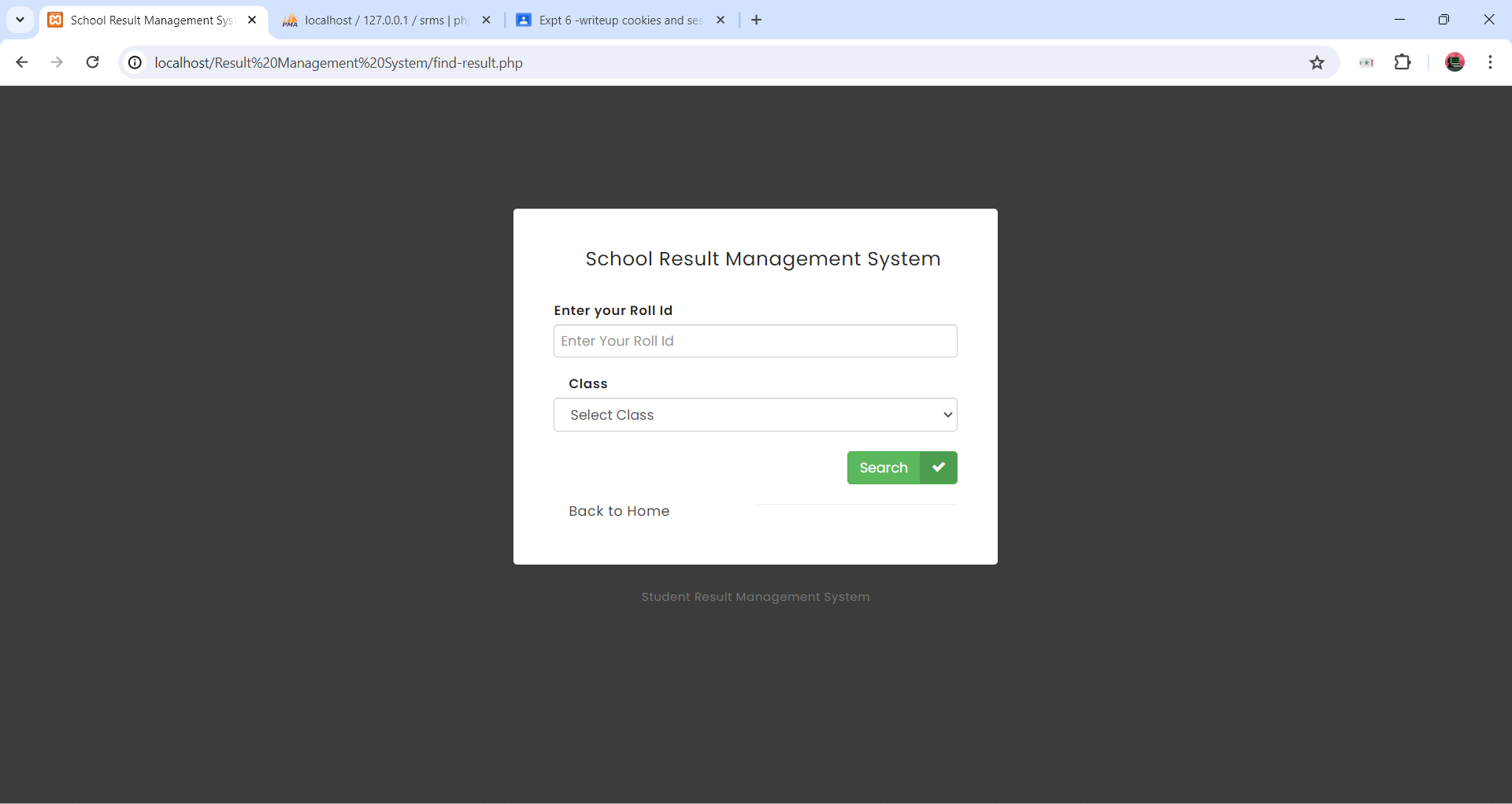
            });

        </script>

        <!-- ========== ADD custom.js FILE BELOW WITH YOUR CHANGES ========== -->

    </body>

</html>

****

**Post Lab Objective with Ans :**

1. Differentiate between sessions and cookies

Ans:

* **Sessions:** Sessions are server-side storage mechanisms used to maintain stateful information about a user's interaction with a web application. When a user visits a website, a unique session ID is generated, which is stored on the server. Session data can include user authentication details, shopping cart contents, or any other information relevant to the user's session. Sessions are typically more secure than cookies because the data is stored on the server, not on the user's device.
* **Cookies:** Cookies are small pieces of data stored on the user's device by the web browser. They are used to store user preferences, track user behavior, and maintain state between different sessions. Unlike sessions, cookies are stored on the client-side, making them susceptible to tampering and security risks. Cookies can be either persistent (stored for a specific duration) or session-based (deleted when the browser is closed).

1. Describe the methods used to properly destroy sessions and cookies in a web application.

Ans:

**Destroying Sessions:** Sessions are usually destroyed by calling the **session\_destroy()** function in PHP. This function removes all session data associated with the current session. Additionally, it's a good practice to unset any session variables that are no longer needed before destroying the session. Example:

<?php

session\_start();

// Unset all session variables

$\_SESSION = array();

// Destroy the session

session\_destroy();

?>

**Destroying Cookies:** Cookies can be destroyed by setting their expiration time to a past date. This will prompt the browser to remove the cookie from its storage. Example:

<?php

// Set the cookie's expiration time to a past date

setcookie("cookie\_name", "", time() - 3600, "/");

?>

1. Discuss strategies for implementing security measures in form sessions

Ans:

* + **Use HTTPS:** Ensure that your web application is served over HTTPS to encrypt data transmitted between the client and the server, including session information.
  + **Session ID Protection:** Protect session IDs from session fixation attacks by regenerating session IDs after successful authentication or periodically throughout the session.
  + **Validate Input Data:** Always validate and sanitize user input to prevent injection attacks such as SQL injection and cross-site scripting (XSS).
  + **Implement CSRF Protection:** Use CSRF tokens to protect against cross-site request forgery attacks by ensuring that each form submission includes a unique token.
  + **Limit Session Lifetime:** Set a reasonable session expiration time to minimize the risk of session hijacking.
  + **Implement Authentication Measures:** Implement strong authentication mechanisms such as multi-factor authentication (MFA) to verify the identity of users accessing the application.
  + **Regular Security Audits:** Conduct regular security audits and vulnerability assessments to identify and address any potential security flaws in the application.
  + **Educate Users:** Educate users about best security practices, such as avoiding public Wi-Fi networks and using strong, unique passwords.

Top of Form