МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ   
ІМЕНІ ВАСИЛЯ СТЕФАНИКА

Кафедра комп’ютерних наук та інформаційних систем

**Лабораторна робота №6**

з курсу «Веб-програмування (РНР)»

на тему:

«Задача аутентифікації. Куки, сесії.»

**Виконав:**

студент групи КН-41

Сенів П.В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 р.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис)

**Перевірив:**

викладач кафедри КНІС

к.т.н., доцент Горєлов В.О. «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 р.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оцінка, підпис)

м. Івано-Франківськ

2021

Мета: набути теоретичних знань на отримати практичні навички розв’язку задачі аутентифікації.

Завдання: ознайомитися із документацією, реалізувати підтримку session, cookie, хешування паролю та збереження даних у базі даних.

Хід роботи

1. Реалізуйте реєстрацію користувача у системі. Передбачте хешування паролю актуальним алгоритмом хешування та введіть поняття ролі (адміністратор, користувач). Не зберігайте у базі даних пароль у відкритому вигляді – тільки хеш.

2. Надайте можливість залогуватися та вилогуватися. Передбачте певний часу збереження даних у cookies (однакового для всіх). Запропонуйте користувачеві запам’ятати його дані для наступного входу без явного логування.

3. Використайте механізм session при переході між сторінками ресурсу. 4. Створіть сторінку, вміст якої буде залежати від ролі.

5. Розгляньте різні сценарії, наприклад:

5.1. користувач – зареєстрований, проте не залогований;

5.2. користувач не зареєстрований;

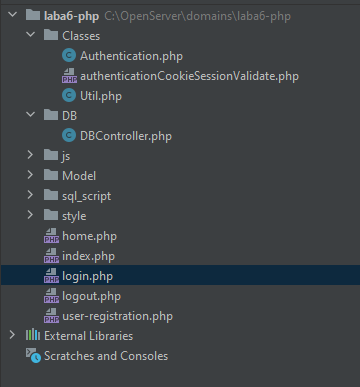
5.3. session – є, cookies – відсутні;

5.4. session – немає, наявні cookies;

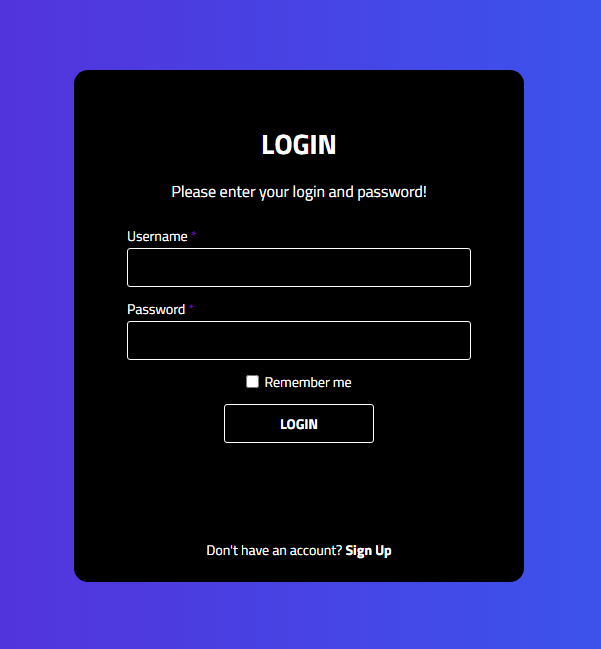
5.5. користувач ввів хибну пару логін/пароль.

6. Оформте звіт.

**Результати виконання роботи:**



**Рисунок 1. Структура проекту.**



**Рисунок 2. Інтерфейс сторінки входу**

**Файл DBController.php**

<?php  
*/\*\*  
 \* Copyright (C) Phppot  
 \*  
 \* Distributed under 'The MIT License (MIT)'  
 \* In essense, you can do commercial use, modify, distribute and private use.  
 \* Though not mandatory, you are requested to attribute Phppot URL in your code or website.  
 \*/*namespace model;  
  
*/\*\*  
 \* Generic datasource class for handling DB operations.  
 \* Uses MySqli and PreparedStatements.  
 \*  
 \** ***@version*** *2.6 - recordCount function added  
 \*/*class DBController  
{  
  
 const *HOST* = 'localhost';  
  
 const *USERNAME* = 'lab\_php';  
  
 const *PASSWORD* = 'F33]mu4b-tXRpsNo';  
  
 const *DATABASENAME* = 'lab\_php';  
  
 private $conn;  
  
 */\*\*  
 \* PHP implicitly takes care of cleanup for default connection types.  
 \* So no need to worry about closing the connection.  
 \*  
 \* Singletons not required in PHP as there is no  
 \* concept of shared memory.  
 \* Every object lives only for a request.  
 \*  
 \* Keeping things simple and that works!  
 \*/* function \_\_construct()  
 {  
 $this->conn = $this->connectDB();  
 }  
  
 */\*\*  
 \* If connection object is needed use this method and get access to it.  
 \* Otherwise, use the below methods for insert / update / etc.  
 \*  
 \** ***@return*** *\mysqli  
 \*/* public function connectDB()  
 {  
 $conn = new \mysqli(self::*HOST*, self::*USERNAME*, self::*PASSWORD*, self::*DATABASENAME*);  
  
 if (mysqli\_connect\_errno()) {  
 trigger\_error("Problem with connecting to database.");  
 }  
  
 $conn->set\_charset("utf8");  
 return $conn;  
 }  
  
  
 function runBaseQuery($query)  
 {  
 $result = mysqli\_query($this->conn, $query);  
 while ($row = mysqli\_fetch\_assoc($result)) {  
 $resultset[] = $row;  
 }  
 if (!empty($resultset))  
 return $resultset;  
 }  
  
  
 function runQuery($query, $param\_type, $param\_value\_array)  
 {  
  
 $sql = $this->conn->prepare($query);  
 $this->bindQueryParams($sql, $param\_type, $param\_value\_array);  
 $sql->execute();  
 $result = $sql->get\_result();  
  
 if ($result->num\_rows > 0) {  
 while ($row = $result->fetch\_assoc()) {  
 $resultset[] = $row;  
 }  
 }  
  
 if (!empty($resultset)) {  
 return $resultset;  
 }  
 }  
  
 */\*\*  
 \* To get database results  
 \*  
 \** ***@param*** *string $query  
 \** ***@param*** *string $paramType  
 \** ***@param*** *array $paramArray  
 \** ***@return*** *array  
 \*/* public function select($query, $paramType = "", $paramArray = array())  
 {  
 $stmt = $this->conn->prepare($query);  
  
 if (!empty($paramType) && !empty($paramArray)) {  
  
 $this->bindQueryParams($stmt, $paramType, $paramArray);  
 }  
 $stmt->execute();  
 $result = $stmt->get\_result();  
  
 if ($result->num\_rows > 0) {  
 while ($row = $result->fetch\_assoc()) {  
 $resultset[] = $row;  
 }  
 }  
  
 if (!empty($resultset)) {  
 return $resultset;  
 }  
 }  
  
 */\*\*  
 \* To insert  
 \*  
 \** ***@param*** *string $query  
 \** ***@param*** *string $paramType  
 \** ***@param*** *array $paramArray  
 \** ***@return*** *int  
 \*/* public function insert($query, $paramType, $paramArray)  
 {  
 $stmt = $this->conn->prepare($query);  
 $this->bindQueryParams($stmt, $paramType, $paramArray);  
  
 $stmt->execute();  
 $insertId = $stmt->insert\_id;  
 return $insertId;  
 }  
  
 */\*\*  
 \* To execute query  
 \*  
 \** ***@param*** *string $query  
 \** ***@param*** *string $paramType  
 \** ***@param*** *array $paramArray  
 \*/* function update($query, $param\_type, $param\_value\_array)  
 {  
 $sql = $this->conn->prepare($query);  
 $this->bindQueryParams($sql, $param\_type, $param\_value\_array);  
 $sql->execute();  
 }  
  
 public function execute($query, $paramType = "", $paramArray = array())  
 {  
 $stmt = $this->conn->prepare($query);  
  
 if (!empty($paramType) && !empty($paramArray)) {  
 $this->bindQueryParams($stmt, $paramType, $paramArray);  
 }  
 $stmt->execute();  
 }  
  
 */\*\*  
 \* 1.  
 \* Prepares parameter binding  
 \* 2. Bind prameters to the sql statement  
 \*  
 \** ***@param*** *string $sql  
 \** ***@param*** *string $paramType  
 \** ***@param*** *array $paramArray  
 \*/* public function bindQueryParams($sql, $paramType, $paramArray = array())  
 {  
 $paramValueReference[] = &$paramType;  
 for ($i = 0; $i < count($paramArray); $i++) {  
 $paramValueReference[] = &$paramArray[$i];  
 }  
 call\_user\_func\_array(array(  
 $sql,  
 'bind\_param'  
 ), $paramValueReference);  
 }  
  
 */\*\*  
 \* To get database results  
 \*  
 \** ***@param*** *string $query  
 \** ***@param*** *string $paramType  
 \** ***@param*** *array $paramArray  
 \** ***@return*** *array  
 \*/* public function getRecordCount($query, $paramType = "", $paramArray = array())  
 {  
 $stmt = $this->conn->prepare($query);  
 if (!empty($paramType) && !empty($paramArray)) {  
  
 $this->bindQueryParams($stmt, $paramType, $paramArray);  
 }  
 $stmt->execute();  
 $stmt->store\_result();  
 $recordCount = $stmt->num\_rows;  
  
 return $recordCount;  
 }  
  
}

**Файл login.php**

<?php  
namespace model;  
  
use model\Member;  
  
session\_start();  
  
require\_once "Classes/Authentication.php";  
require\_once "Classes/Util.php";  
  
$auth = new Authentication();  
$db\_handle = new DBController();  
$util = new Util();  
  
require\_once "Classes/authenticationCookieSessionValidate.php";  
  
if ($isLoggedIn) {  
 $util->redirect("home.php");  
}  
  
if (!empty($\_POST["login"])) {  
  
 $isAuthenticated = false;  
  
 $username = $\_POST["member\_name"];  
 $password = $\_POST["member\_password"];  
 $user = $auth->getMemberByUsername($username);  
 if (password\_verify($password, $user[0]["member\_password"])) {  
 $isAuthenticated = true;  
 }  
  
 if ($isAuthenticated) {  
  
 $\_SESSION["member\_name"] = $user[0]["member\_name"]; //Запис у сесію ім'я  
  
 $\_SESSION["member\_id"] = $user[0]["member\_id"];  
  
 // Set Authentication Cookies if 'Remember Me' checked  
 if (!empty($\_POST["remember"])) {  
 setcookie("member\_login", $username, $cookie\_expiration\_time);  
  
 $random\_password = $util->getToken(16);  
 setcookie("random\_password", $random\_password, $cookie\_expiration\_time);  
  
 $random\_selector = $util->getToken(32);  
 setcookie("random\_selector", $random\_selector, $cookie\_expiration\_time);  
  
 $random\_password\_hash = password\_hash($random\_password, *PASSWORD\_DEFAULT*);  
 $random\_selector\_hash = password\_hash($random\_selector, *PASSWORD\_DEFAULT*);  
  
 $expiry\_date = date("Y-m-d H:i:s", $cookie\_expiration\_time);  
  
 // mark existing token as expired  
 $userToken = $auth->getTokenByUsername($username, 0);  
 if (!empty($userToken[0]["id"])) {  
 $auth->markAsExpired($userToken[0]["id"]);  
 }  
 // Insert new token  
 $auth->insertToken($username, $random\_password\_hash, $random\_selector\_hash, $expiry\_date);  
 } else {  
 $util->clearAuthCookie();  
 }  
  
 session\_write\_close();  
 $util->redirect("home.php");  
 } else {  
 $message = "Invalid username or password";  
 }  
}  
  
?>  
<!doctype html>  
<html lang="uk">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <meta name="author" content="Seniv Pavlo, https://github.com/PavloSeniv">  
 <meta name="copyright" content="Seniv Pavlo">  
 <link rel="preconnect" href="https://fonts.googleapis.com">  
 <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>  
 <link href="https://fonts.googleapis.com/css2?family=Titillium+Web:wght@300;400;700&display=swap"  
 rel="stylesheet">  
 <title>Login</title>  
 <!-- ======== Style ======== -->  
 <link rel="stylesheet" href="style/style.css">  
  
 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>  
</head>  
<body class="container">  
<div class="phppot-container">  
 <div class="sign-up-container">  
  
 <div class="signup-align">  
 <form name="login" action="" id="frmLogin" method="post"  
 onsubmit="return loginValidation()">  
  
 <div class="signup-heading">Login</div>  
  
 <div class="signup-subheading"> Please enter your login and password!</div>  
  
 <div class="error-message"><?php if (isset($message)) {  
 echo $message;  
 } ?></div>  
  
 <div class="row">  
 <div class="inline-block">  
 <div class="form-label">  
 Username<span class="required error" id="username-info"></span>  
 </div>  
  
 <input name="member\_name" type="text" id="username"  
 value="<?php if (isset($\_COOKIE["member\_login"])) {  
 echo $\_COOKIE["member\_login"];  
 } ?>"  
 class="input-field">  
 </div>  
 </div>  
 <div class="row">  
 <div class="inline-block">  
 <div class="form-label">  
 Password<span class="required error" id="login-password-info"></span>  
 </div>  
 <input name="member\_password" type="password" id="login-password"  
 value="<?php if (isset($\_COOKIE["member\_password"])) {  
 echo $\_COOKIE["member\_password"];  
 } ?>"  
 class="input-field">  
 </div>  
 </div>  
  
 <div class="row">  
 <div class="checkbox-block">  
 <input type="checkbox" name="remember" id="remember"  
 <?php if (isset($\_COOKIE["member\_login"])) { ?> checked  
 <?php } ?>  
 />  
 <label for="remember-me">Remember me</label>  
 </div>  
 </div>  
 <div class="row">  
 <input type="submit" name="login" value="Login" id="login-btn"  
 class="form-submit-button">  
 </div>  
 <div class="row">  
 <div class="login-signup">  
 <p>  
 Don't have an account?  
 <a href="user-registration.php">Sign Up</a>  
 </p>  
 </div>  
 </div>  
 </form>  
  
 </div>  
 </div>  
</div>  
  
<script>  
 function loginValidation() {  
 var valid = true;  
 $("#username").removeClass("error-field");  
 $("#password").removeClass("error-field");  
  
 var UserName = $("#username").val();  
 var Password = $('#login-password').val();  
  
 $("#username-info").html("").hide();  
  
 if (UserName.trim() == "") {  
 $("#username-info").html("required.").css("color", "#6a11cb").show();  
 $("#username").addClass("error-field");  
 valid = false;  
 }  
 if (Password.trim() == "") {  
 $("#login-password-info").html("required.").css("color", "#6a11cb").show();  
 $("#login-password").addClass("error-field");  
 valid = false;  
 }  
 if (valid == false) {  
 $('.error-field').first().focus();  
 valid = false;  
 }  
 return valid;  
 }  
</script>  
</body>  
</html>

**Файл home.php**

<?php  
  
use model\Member;  
  
session\_start();  
  
require\_once "Classes/authenticationCookieSessionValidate.php";  
  
if (isset($\_SESSION["member\_name"])) {  
 $username = $\_SESSION["member\_name"];  
 session\_write\_close();  
} else {  
 // since the username is not set in session, the user is not-logged-in  
 // he is trying to access this page unauthorized  
 // so let's clear all session variables and redirect him to index  
 session\_unset();  
 session\_write\_close();  
  
 $url = "./index.php";  
 header("Location: $url");  
}  
  
?>  
<!doctype html>  
<html lang="uk">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <meta name="author" content="Seniv Pavlo, https://github.com/PavloSeniv">  
 <meta name="copyright" content="Seniv Pavlo">  
 <link rel="preconnect" href="https://fonts.googleapis.com">  
 <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>  
 <link href="https://fonts.googleapis.com/css2?family=Titillium+Web:wght@300;400;700&display=swap"  
 rel="stylesheet">  
 <title>Home</title>  
 <!-- ======== Style ======== -->  
 <link rel="stylesheet" href="style/style.css">  
  
</head>  
<body>  
<div>  
  
 <main id="Main" class="phppot-container">  
 <div class="home">  
 <div class="page-header">  
 <p class="login-signup-home"><a href="logout.php">Logout</a></p>  
 </div>  
 <div class="page-content">Welcome  
 <?php if ($username == "admin") {  
 echo "<span class='username-admin'>";  
 echo $username;  
 echo "</span>";  
 } else {  
 echo "<span class='username-member'>";  
 echo $username;  
 echo "</span>";  
 }  
 ?></div>  
 </div>  
 </main>  
</div>  
  
<script src="js/three.min.js"></script>  
<script src="js/vanta.halo.min.js"></script>  
<script src="js/Vanta.js"></script>  
</body>  
</html>

**Висновок:** на лабораторній роботі набув теоретичних знань та отримав практичні навички розв’язку задачі аутентифікації. Реалізовано підтримку session, cookie, хешування паролю та збереження даних у базі даних. Додано валідацію даних, як на стороні клієнта так і на стороні сервера. Додано збереження та видалення cookie при виході з акаунту.