**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**імені ТАРАСА ШЕВЧЕНКА**

****

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра прикладних інформаційних систем**

**Звіт до практичної роботи №3**

**з курсу**

**«Бази Даних»**

*студента 2 курсу*

*групи ПП-22*

*спеціальності 122 «Комп'ютерні науки»*

*ОП «Прикладне програмування»*

Шевлюк Вікторії Віталіївни

*Викладач:*

асистент

Криволапов Я. В.

**Київ – 2022**

**Тема:** Системи ідентифікація та верифікація входу до бази даних.

**Мета:** Набуття навичок створення простих серверних скриптів для реєстрації, ідентифікації та верифікації користувача.

**Завдання:**

1. Створити в базі даних таблицю для реєстрації користувачів;
2. Написати мовою PHP скрипт для реєстрації користувачів в базі даних;
3. Написати мовою PHP скрипт, який надавав би користувачам початкові Привілеї, необхідні для виконання операцій переляду, зміни та додавання даних до бази даних;
4. Написати мовою PHP скрипт, який виконував би верифікацію користувачів при вході до бази дних.

**Хід роботи:**

Спочатку нам потрібно створити в нашій базі даних таблицю, що буде зберігати дані користувача:



В цій таблиці я створила поля для ім’я користувача, айді користувача, паролю, хешування, поштової адреси, посилання та номеру телефону.

Далі створимо скрипт, в якому реалізуємо графічний інтерфейс з полями для заповнення (для логіну) та з посиланням на реєстрацію. Це буде форма входу користувача у систему.

Ось так виглядає код скрипту:

<form action="Lab3\_2.php" method="post">

Login:<br>

<input type="text" name="login"><br>

Password:<br>

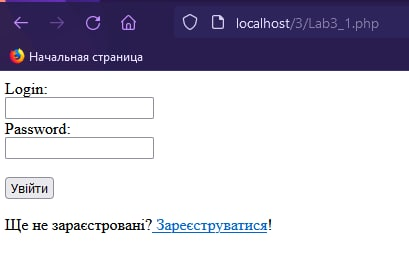
<input type="password" name="password"><br><br>

<input type="submit" value="Увійти">

<p class="regtext">Ще не зараєстровані?<a href = "reg.php"> Зареєструватися</a>!</p>

</form>

Якщо ми викличемо цей файл у браузері, отримаємо наступний результат:



Тепер нам потрібно написати скрипт що буде реєструвати нового користувача у систему і який надавав би користувачам початкові Привілеї, необхідні для виконання операцій переляду, зміни та додавання даних до бази даних. У цьому скрипті відбувається реалізація графічного інтерфейсу з полями, що відповідатимуть полям нашої таблиці. Також там є обов’язкові перевірки на коректне введення інформації від користувача у поля, і якщо інформація проходить перевірки, то вона передається далі на обробку наступник файлом.

Код даного скрипту:

<form method="post">

<table>

<tr><td>Ім'я</td><td><input type="text" name='login'></td></tr>

<tr><td>Пароль</td><td><input type="password" name="pass"></td></tr>

<tr><td>Повторіть пароль</td><td><input type="password" name="pass\_rep"></td></tr>

<tr><td>e-mail</td><td><input type="text" name="email"></td></tr>

<tr><td>URL</td><td><input type="text" name="url"></td></tr>

<tr><td>Телефон</td><td><input type="text" name="tel"></td></tr>

<tr><td> </td><td><input type="submit" value="Реєстрація"></td></tr>

</table>

</form>

<?php

session\_start();

// session.gc\_maxlifetime=1440

if(empty ($\_POST)) exit();

$\_POST['login'] = trim($\_POST['login']);

$\_POST['pass'] = trim($\_POST['pass']);

$\_POST['pass\_rep'] = trim($\_POST['pass\_rep']);

if(empty ($\_POST['login'])) exit('Поле "Ім\'я" не заповнено');

if(empty ($\_POST['pass'])) exit('Введіть "Пароль"');

if(empty ($\_POST['pass\_rep'])) exit('Введіть пароль ще раз');

if($\_POST['pass'] != $\_POST['pass\_rep']) exit('Пароль та підтвердження не співпадають');

if(!empty ($\_POST['email']))

{

if(!preg\_match("|^[-0-9a-z\_]+@[-0-9a-z\_]+\.[a-z]{2,6}$|i",$\_POST['email']))

exit('Поле "e-mail" повинно відповідати формату name@ukr.net');

}

if(!empty ($\_POST['tel']))

{

if(!preg\_match("|^[0-9]{10}$|",$\_POST['tel']))

exit('Поле "Телефон" заповнено не правильно');

}

include "config.php";

$\_POST['login']=mysqli\_escape\_string($dbcon,$\_POST['login']);

$\_POST['pass']=mysqli\_escape\_string($dbcon,$\_POST['pass']);

$\_POST['email']=mysqli\_escape\_string($dbcon,$\_POST['email']);

$\_POST['url']=mysqli\_escape\_string($dbcon,$\_POST['url']);

$\_SESSION['login']=$\_POST['login'];

$\_SESSION['pass']=$\_POST['pass'];

$\_SESSION['email']=$\_POST['email'];

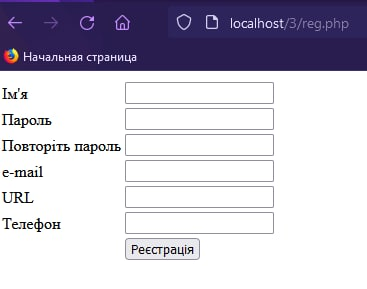
$\_SESSION['url']=$\_POST['url'];

$\_SESSION['tel']=$\_POST['tel'];

include "config2.php";

?>

В результаті отримуємо таку форму:



Наступник скрипт буде обробляти інформацію, занесену користувачем, і додавати її у таблицю, що ми створили на початку цієї роботи. Так ми створюємо облікові записи користувачів.

Так виглядає код скрипту:

<?php

$query= "SELECT COUNT(\*) FROM userlist WHERE USER\_ID = '$\_SESSION[login]'";

$urs=mysqli\_query($dbcon,$query);

if(!$urs) exit("Помилка - ".mysqli\_error($dbcon));

$rez\_total=mysqli\_fetch\_row($urs);

$total=$rez\_total[0];

if($total>0) exit("Користувач з ім'ям <b>".$\_SESSION['login']."</b> вже зареэстрований");

$str='0-Te=d7Z';

$str\_hach = password\_hash($\_SESSION['pass'],PASSWORD\_DEFAULT);

$str\_md5 = MD5($\_SESSION['pass']);

$str\_md5\_salt = MD5($\_SESSION['pass'].$str);

$query2 = "INSERT INTO userlist

VALUES (null, '".$\_SESSION['login']."', '".$\_SESSION['pass']."', '".$str\_hach."', '".$str\_md5."', '".$str\_md5\_salt."', '".$\_SESSION['email']."', '".$\_SESSION['url']."', '".$\_SESSION['tel']."')";

$query3 = "CREATE USER ".$\_SESSION['login']." IDENTIFIED BY '".$\_SESSION['pass']."'";

if(mysqli\_query($dbcon, $query3))

{

echo "<P>Реєстрація пройшла успішно</P>";

} else

{

echo "<P>Виник збій при реєстрації #1</P>";

echo mysqli\_error($dbcon);

}

if(mysqli\_query($dbcon, $query2)){

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

echo "<HTML><HEAD>

<META HTTP-EQUIV='Refresh' CONTENT='0; URL=$\_SERVER[PHP\_SELF]'>

</HEAD></HTML>";

session\_destroy();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

$query4 = "GRANT SELECT, INSERT, UPDATE ON viktoriadb.\* TO ".$\_SESSION['login'];

$query4 = "GRANT ALL ON viktoriadb.\* TO ".$\_SESSION['login'];

if(!mysqli\_query($dbcon,$query4))

{

echo "<P>Привілеї не встановлено</P>";

exit("Помилка - ".mysqli\_error($dbcon));

}

echo"

<table>

<form action='end.php' method='post'>

<p><input type='submit' value='Обнулити сесійні значення'></p>

</form>

</table>";

}

else

{

echo "<P>Виник збій при реєстрації #2</P>";

echo mysqli\_error($dbcon);

}

$query5 = "show GRANTS FOR '".$\_SESSION['login']."'@'%'";

echo $query5;

echo "<br>";

$ver=mysqli\_query($dbcon,$query5);

if(!$ver)

{

echo "<P> Не вдалося показати привілеї </P>";

exit(mysqli\_error($dbcon));

}

echo "<table border=2>";

while(list($id)=mysqli\_fetch\_row($ver))

{

echo "<tr>

<td> $id</td>

</tr>";

}

echo "</table>";

if(!mysqli\_close($dbcon))

{

echo "Зв'язок з хостом ".$dblocation." не розірвано";

exit("Помилка - ".mysqli\_error($dbcon));

} else

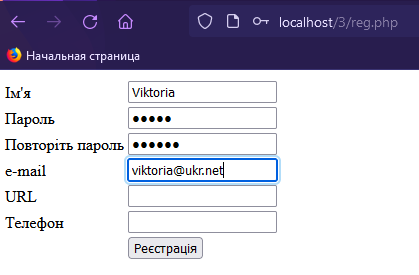
{

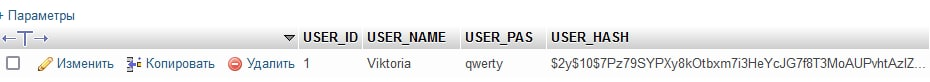
echo "Зв'язок з хостом ".$dblocation." розірвано";

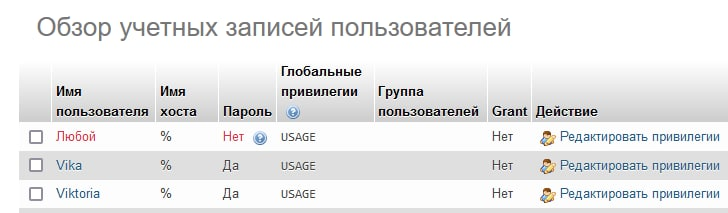
}

?>

Тепер коли ми вносимо дані в поля реєстрації та натискаємо кнопку «Реєстрація», отримуємо наступний результат:







Далі напишемо скрипт, що буде проводити верифікацію користувача. Після проходження верифікації можна отримати доступ до свого «профілю» і до інших даних (в залежності від статусу користувача)

Код скрипта:

<?php

session\_start();

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

{

$\_SESSION['login'] = $\_POST['login'];

$\_SESSION['password'] = $\_POST['password'];

}

require\_once("config3.php");

if(empty($\_GET['USER\_ID']))

{

echo "<br>";

$query= "SELECT \* FROM userlist ORDER BY USER\_NAME";

$usr=mysqli\_query($dbcon,$query);

if(!$usr) exit ("Помилка - ".mysqli\_error($dbcon));

while($user=mysqli\_fetch\_array($usr, MYSQLI\_BOTH))

{

echo "<a href=$\_SERVER[PHP\_SELF]?USER\_ID=$user[USER\_ID]> $user[USER\_NAME]</a><br>";

}

echo "<a href= 'Lab3\_1.php'>На початок</a><br>";

}

else

{

echo "<br>";

$query="SELECT \* FROM userlist WHERE USER\_ID=$\_GET[USER\_ID]";

$usr=mysqli\_query($dbcon,$query);

if(!$usr) exit ("Помилка - ".mysqli\_error($dbcon));

$user=mysqli\_fetch\_array($usr,MYSQLI\_BOTH);

echo "Ім'я користувача - ".$user['USER\_NAME']."<br>";

if(!empty($user['EMAIL'])) echo "e-mail - ".$user['EMAIL']."<br>";

if(!empty($user['URL'])) echo "URL - ".$user['URL']."<br>";

echo "<br>";

echo "<br>";

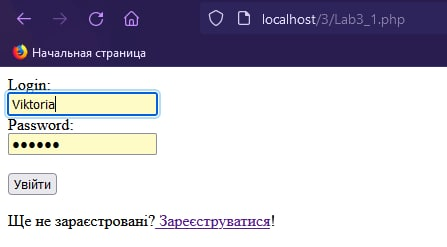
echo "<a href= 'Lab3\_2.php'>Повернутися</a><br>";

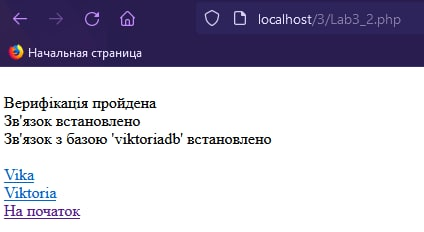
echo "<a href= 'Lab3\_1.php'>На початок</a><br>";

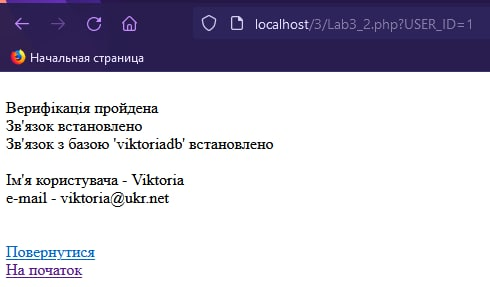
}

?>

Ось результат виконання наших скриптів:







**Висновок:** У ході даної лабораторної роботия набула навичок створення простих серверних скриптів для реєстрації, ідентифікації та верифікації користувача.

**Контрольні питання:**

1. ***Чим відрізняються ідентифікація та верифікація?***

Індентифікація – знаходження даних невідомої особи через порівняння шаблонів, що ісували у системі.

Верифікація – перевірка чи співпадає введена інформація конкретному шаблону для користувача.

1. ***Яке призначення функції password\_hash()?***

Дана функція спеціальним алгоритмом створює хеш пароля.

1. ***Що таке «сіль» хеш-коду і навіщо вона потрібна?***

Сіль хеш-коду це спеціальний набір символів що використовується для ускладнення хешу, що робить таке шифрування більш надійним і стійким.

1. ***Що таке сесія?***

Сессія – певний набір даних, що відносяться лише користувача і зберігаються на сервері.

1. ***Що таке методи POST та GET?***

Метод GET надсилає закодовану інформацію користувача, додану до запиту сторінки. Сторінка та закодована інформація розділені символом «?».

Метод POST передає інформацію через заголовки HTTP. Інформація кодується, як описано у випадку методу GET, і поміщається в заголовок під назвою QUERY\_STRING.

1. ***Які існують варіанти надання привілеїв?***

ALL PRIVILEGES – користувач отримує всі привілеї на визначеному рівні доступу.

CREATE – користувач отримує дозвіл на створення баз даних і таблиць.

DROP – користувач отримує дозвіл на видалення баз даних і таблиць.

DELETE – користувач отримує дозвіл на видалення рядків із певної таблиці.

INSERT – користувач отримує дозвіл вставляти рядки в певну таблицю.

SELECT – користувач отримує дозвіл на читання бази даних.

UPDATE – користувач отримує дозвіл оновлювати рядки таблиці.

1. ***Яке призначення має оператор GRANT?***

Оператор використовується для призначення або надання привілеїв користувачам.