**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет   
імені Петра Могили**

**Факультет комп’ютерних наук**

**Кафедра інтелектуальних інформаційних систем**

**ЗВІТ**

*з лабораторної роботи № 2*

Дисципліна «П ІС та ТЗІ»

Спеціальність 122 «Комп'ютерні науки»

122 – ЛР.ПЗ.02 – 402.1610203

Виконала: студентка групи 402

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К. О. Бондаренко\_\_\_\_

(підпис, ініціали та прізвище)

\_\_\_\_\_\_\_\_\_\_\_\_ 22.09.2019\_\_\_\_\_\_\_

(дата)

Перевірив:

\_\_\_\_\_\_\_\_\_О. Р. Тогоєв \_\_\_\_\_

(підпис, ініціали та прізвище)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_22.09.2019\_\_\_\_\_\_\_\_\_\_

(дата)

**Миколаїв – 2019**

**Лабораторна робота №2**

**Завдання для студентів:**

1. Розробити форму реєстрації з записом у БД за допомогою JS/PHP.
2. Зашифрувати дані форми MD5.
3. Оформити звіт.

**Теоретичні відомості:**

**MD5**

MD5 - 128-бітний алгоритм хешування, розроблений професором Рональдом Л. Ривестом з Массачусетського технологічного інституту (Massachusetts Institute of Technology, MIT) в 1991 році. Призначений для створення «відбитків» або дайджестів повідомлення довільної довжини і подальшої перевірки їх достовірності. Широко застосовувався для перевірки цілісності інформації та зберігання хешів паролів.

На вхід алгоритму надходить вхідний потік даних, хеш якого необхідно знайти. Довжина повідомлення вимірюється в бітах і може бути будь-який (в тому числі нульовою). Запишемо довжину повідомлення в L. Це число ціле і невід'ємне. Кратність будь-яким числам необов'язкова. Після надходження даних йде процес підготовки потоку до обчислень.

Хеш-функція призначена для згортки вхідного масиву будь-якого розміру в бітову рядок, для MD5 довжина вихідний рядки дорівнює 128 бітам. Для чого це потрібно? Наприклад у вас є два масиви, а вам необхідно швидко порівняти їх на рівність, то хеш-функція може зробити це за вас, якщо у двох масивів хеші різні, то масиви гарантовано різні, а в разі рівності хеш - масиви швидше за все рівні.  
Однак найчастіше хеш-функції використовуються для перевірки унікальності пароля, файлу, рядка і тд. Наприклад, завантажуючи файл з інтернету, ви часто бачите поруч з ним рядок виду b10a8db164e0754105b7a99be72e3fe5 - це і є хеш, прогнавши цей файл через алгоритм MD5 ви отримаєте такий рядок, і, якщо хеши рівні, можна з великою ймовірністю стверджувати що цей файл дійсно справжній (звичайно з деякими застереженнями, про які розповім далі).  
**Алгоритм складається з п'яти кроків:**  
1) Append Padding Bits  
У вихідний рядок дописують одиничний байт 0х80, а потім дописують нульові біти, до тих пір, поки довжина повідомлення не буде порівнянна з 448 по модулю 512. Тобто дописуємо нулі до тих пір, поки довжина нового повідомлення не буде дорівнює [довжина] = ( 512 \* N + 448),  
де N - будь-яке натуральне число, таке, що цей вислів буде найближче до довжини блоку.  
2) Append Length  
Далі в повідомлення дописується 64-бітове представлення довжини вихідного повідомлення.  
3) Initialize MD Buffer  
На цьому кроці инициализируется буффер  
word A 01 23 45 67  
word B: 89 ab cd ef  
word C: fe dc ba 98  
word D: 76 54 32 10  
Як можна помітити буффер складається з чотирьох констант, призначений для збору хеша.

4) Process Message in 16-Word Blocks  
На четвертому кроці в першу чергу визначається 4 допоміжні логічні функції, які перетворять вхідні 32-бітові слова, в, як не дивно, в 32-бітові вихідні.  
F (X, Y, Z) = XY v not (X) Z  
G (X, Y, Z) = XZ v Y not (Z)  
H (X, Y, Z) = X xor Y xor Z  
I (X, Y, Z) = Y xor (X v not (Z))  
Також на цьому етапі реалізується так званий «білий шум» - посилення алгоритму, що складається 64 елементного масиву, що містить псевдовипадкові числа, залежні від синуса числа i:  
T [i] = 4,294,967,296 \* abs (sin (i))  
Далі починається «магія». Копіюємо кожен 16-бітний блок в масив X [16] і виробляємо маніпуляції:  
AA = A  
BB = B  
CC = C  
DD = D  
Потім відбуваються «чудесні» перетворення-раунди, яких всього буде 4. Кожен раунд складається з 16 елементарних перетворень, які в загальному вигляді можна представити у вигляді [abcd ksi], яке, в свою чергу, можна уявити як A = B + (( A + F (B, C, D) + X [k] + T [i]) <<< s), де  
A, B, C, D - регістри  
F (B, C, D) - одна з логічних функцій  
X [k] - k-тий елемент 16-бітного блоку.  
T [i] - i-тий елемент таблиці «білого шуму»  
<<< s - операція циклічного зсуву на s позицій вліво.  
Наводити всі раунди не має сенсу, все їх можна подивитися тут  
Ну і в кінці підсумовуємо результати обчислень:  
A = A + AA  
B = B + BB  
C = C + CC  
D = D + DD  
5) Output  
Виводячи побайтово буффер ABCD починаючи з A і закінчуючи D отримаємо наш хеш.

**Хід виконання роботи**

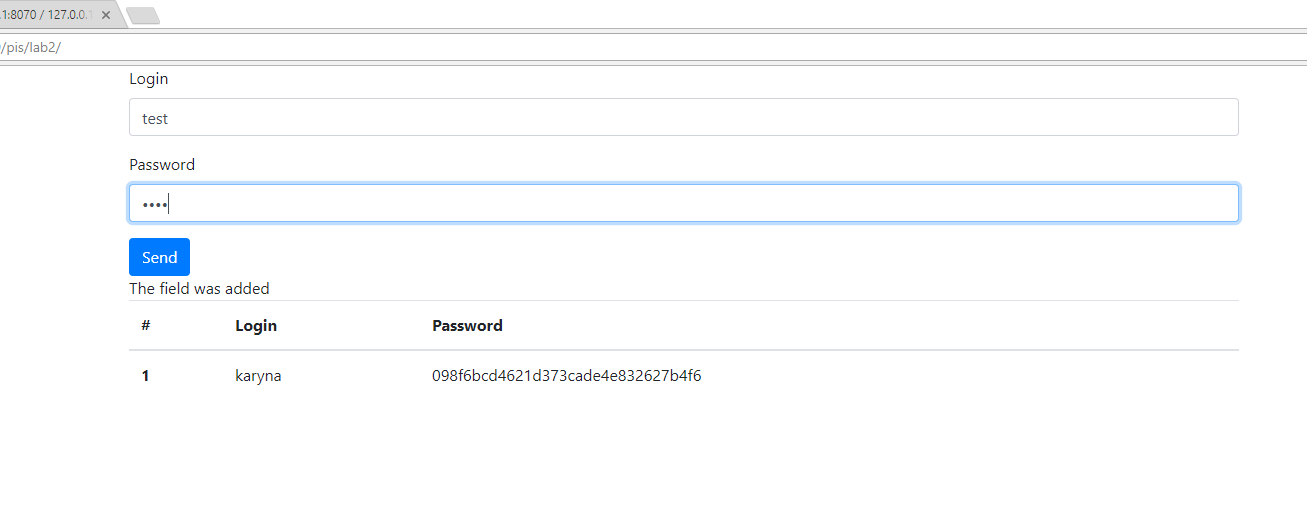
1. На головній сторінці потрібно ввести логін, пароль та натиснути кнопку Send (рис.1).

Рисунок 1 – Головна сторінка.

1. З’явиться в таблиці нижче логін та зашифрований пароль(рис.2).

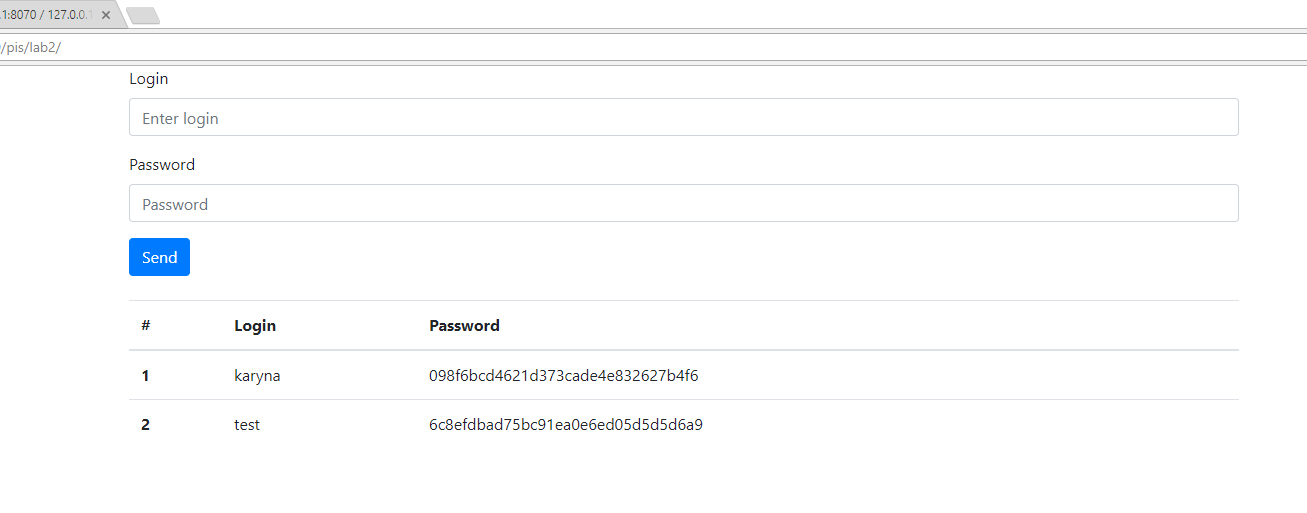


Рисунок 2 – Таблиця з користувачами.

1. Таблиця в базі даних (рис. 3).
2. Рисунок 1 – Головна сторінка.
3. Рисунок 1 – Головна сторінка.

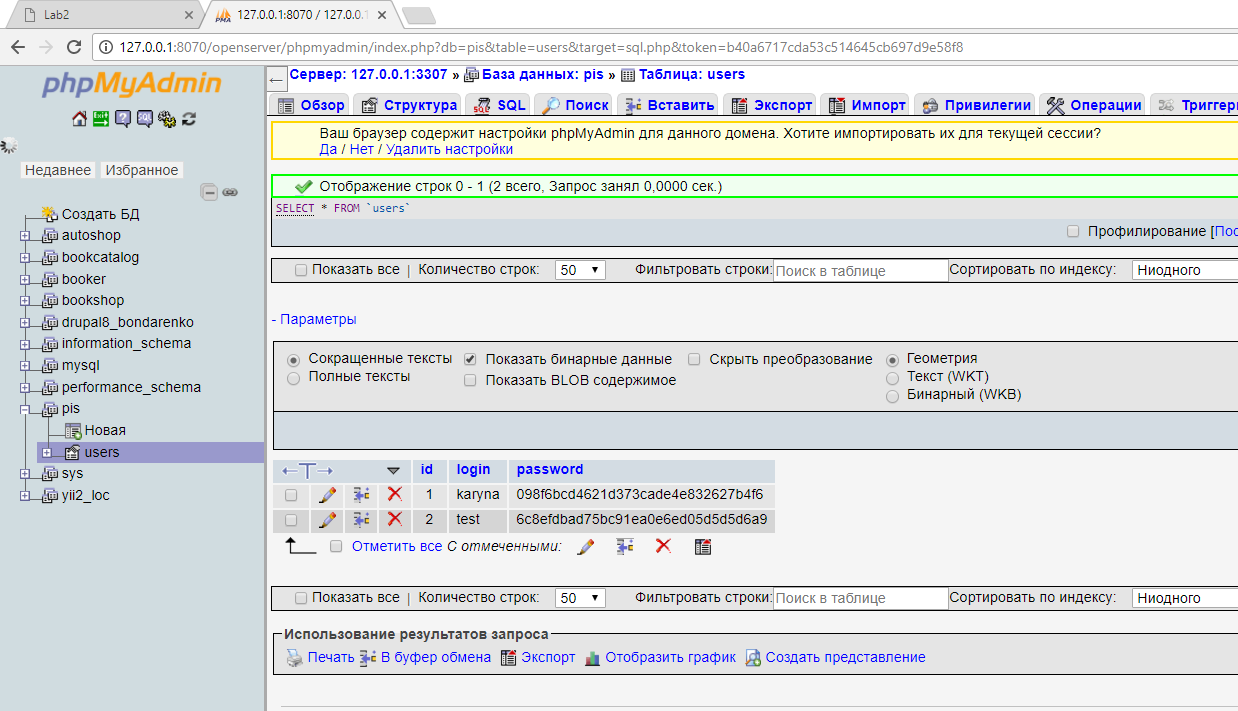


Рисунок 3 – Таблиця user.

**Реалізація**

index.php

**<?php  
  
include\_once 'config.php'**;  
  
**include\_once 'lib/MySql.php'**;  
**include\_once 'lib/Sql.php'**;  
  
$mySql = **new** MySql();  
  
**if**(**isset**($\_POST[**'btn'**]) && $\_POST[**'login'**] && $\_POST[**'password'**])  
{  
 $insert = $mySql->setTable(**'users'**)->setValues($\_POST[**'login'**])->setValues($\_POST[**'password'**])->insert();  
}  
  
$users = $mySql->setTable(**'users'**)->select();  
  
  
**include\_once 'templates/index.php'**;

config.php

**<?php**define(**'HOST'**,**'localhost'**);  
define(**'PORT'**, **'3307'**);  
define(**'USER\_NAME'**,**'root'**);  
define(**'USER\_PASS'**,**''**);  
define(**'DATABASE'**,**'pis'**);  
define(**'TABLE'**,**'users'**);  
**?>**

templates/index.php

**<?php  
?>**<!doctype html>  
<html lang="en">  
<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">  
 <title>Lab1</title>  
 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"  
 integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">  
 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"  
 integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"  
 crossorigin="anonymous"></script>  
</head>  
<body>  
<div class="container">  
 <form method="post">  
 <div class="form-group">  
 <label for="login">Login</label>  
 <input type="text" class="form-control" id="login" name="login" placeholder="Enter login">  
 </div>  
 <div class="form-group">  
 <label for="password">Password</label>  
 <input type="password" class="form-control" name="password" id="password" placeholder="Password">  
 </div>  
 <button type="submit" name="btn" class="btn btn-primary">Send</button>  
 </form>  
 **<?php  
 echo** $insert.**'<br>'**;  
 **?>** <table class="table">  
 <thead>  
 <tr>  
 <th scope="col">#</th>  
 <th scope="col">Login</th>  
 <th scope="col">Password</th>  
 </tr>  
 </thead>  
 <tbody>  
 **<? foreach**($users **as** $user){**?>** <tr>  
 <th scope="row">**<?=**$user[**'id'**]**?>**</th>  
 <td>**<?=**$user[**'login'**]**?>**</td>  
 <td>**<?=**$user[**'password'**]**?>**</td>  
 </tr>  
 **<?**}**?>** </tbody>  
 </table>  
</div>  
  
</body>  
</html>

lib/Sql.php

**<?php  
  
class** Sql  
{  
 **protected** $fields;  
 **protected** $table;  
 **protected** $values;  
 **protected** $querySelect;  
 **protected** $queryInsert;  
  
 **function** \_\_construct()  
 {  
 $this->fields = **array**();  
 $this->values = **array**();  
 $this->table = **""**;  
 }  
  
  
 **public function** setFields($field)  
 {  
 **if** ($field != **"\*"** && $field != **""**) {  
 array\_push($this->fields, $field);  
 **return** $this;  
 } **else** {  
 **return false**;  
 }  
 }  
  
 **public function** setValues($value)  
 {  
 **if** ($value != **""**) {  
 array\_push($this->values, $value);  
 **return** $this;  
 } **else** {  
 **return false**;  
 }  
 }  
  
 **public function** getValues()  
 {  
 **return** $this->values;  
 }  
  
 **public function** setTable($table)  
 {  
 $this->table = $table;  
 **return** $this;  
 }  
  
 **public function** getTable()  
 {  
 **return** $this->table;  
 }  
  
  
 **public function** select()  
 {  
 $this->querySelect = **"SELECT id, login, password FROM "** . $this->table;  
 }  
  
 **function** insert()  
 {  
 $this->queryInsert = **"INSERT"** . **" INTO "** . $this->table . **"(login, password) "** . **"VALUES (:login, :password);"**;  
 }  
  
}  
  
**?>**

lib/MySql.php

**<?php  
include\_once 'Sql.php'**;  
**include\_once 'config.php'**;  
  
**class** MySql **extends** Sql  
{  
 **protected** $connection;  
 **protected** $obj;  
  
 **function** \_\_construct()  
 {  
 **parent**::*\_\_construct*();  
 $this->connection = **new** PDO(**"mysql:host="** . *HOST* . **";port="** . *PORT* . **";dbname="** . *DATABASE* /\*. ";charset=utf8\_unicode\_ci",\*/, *USER\_NAME*, *USER\_PASS*);  
 $this->connection->setAttribute(PDO::*ATTR\_ERRMODE*, PDO::*ERRMODE\_EXCEPTION*);  
 }  
  
 **public function** select()  
 {  
 **parent**::*select*();  
 $result = **array**();  
 $select = $this->connection->prepare($this->querySelect);  
  
 $select->execute();  
 $index = 0;  
 **while** ($row = $select->fetch(PDO::*FETCH\_ASSOC*)) {  
 $result[$index] = $row;  
 $index++;  
 }  
 **return** $result;  
 }  
  
 **function** insert()  
 {  
 **parent**::*insert*();  
 $result = **""**;  
 $insert = $this->connection->prepare($this->queryInsert);  
 $insert->bindParam(**':login'**, $this->values[0]);  
 $insert->bindParam(**':password'**, md5($this->values[1]));  
 **if** ($insert->execute()) {  
 $result = **"The field was added"**;  
 } **else** {  
 $result = **"The field was NOT added"**;  
 }  
 **return** $result;  
 }  
  
}  
  
**?>**

**Висновки**: під час даної лабораторної роботи за допомогою PHP було розроблено сторінку з полями вводу логіна та пароля, що перетворює пароль у хеш за допомогою алгоритму MD5. Були закрыплены навички роботи з OpenServer, PhpMyAdmin та PhpStorm.