МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Севастопольский государственный университет»

кафедра Информационные системы

Гусев Владислав Яковлевич

Институт информационных технологий и управления в технических системах

курс 3 группа ГИС/б-32-о

09.03.02 Информационные системы и технологии (уровень бакалавриата)

ОТЧЕТ

по лабораторной работе №3

по дисциплине: «Веб-технологии»

по теме: «Исследование механизма сессий в PHP»

Отметка о зачете \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

ст. пр. Забаштанский А.К.

(должность) (подпись) (инициалы, фамилия)

Севастополь

2019

1. ЦЕЛЬ РАБОТЫ

Изучить возможности авторизации пользователей с использованием механизма сессий PHP, приобрести практические навыки использования переменных, сохраняющих свое значение при переходе от одной страницы сайта к другой.

1. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. В соответствии с рекоммендациями п. 3.2.7 реализовать зону администрирования сайта и переместить в нее страницы «Редактор блога» и «Загрузка сообщений гостевой книги».

2. Реализовать накопление информации о посетителях(см п. 3.2.6) страниц пользовательского раздела сайта в разработанной для этого таб-лице базы данных. Структура информации:

· Дата и время посещения;

· Web-страница посещения;

· IP-адрес компьютера посетителя;

· Имя хоста компьютера посетителя;

· Название браузера, который использовал посетитель.

3. Добавить в зону администрирования сайта страницу «Статистика по-сещений», отображающую информацию о посетителях сайта.

4. Добавить в зону администратора форму входа, содержащую поля ло-гин и пароль. Логин и пароль админстратора хранить к коде программы. Проверку пароля осуществлять сравнением зашифрованного пароля с вычисленным заранее хешем верного пароля.

5. На всех страницах зоны администратора реализовать проверку авто-ризации и в случае, если посетитель не авторизован как администратор –перенаправлять пользователя на страницу авторизации). Для этой цели возможно добавить в базовый контроллер раздела администратора метод аутентификации, который будет проверять установленную контроллером переменную сессии isAdmin

Добавив вызов данного метода вначале каждого метода контролле-ров раздела администратора, обеспечим защиту раздела администратора от несанкционированного доступа.

6. В пользовательском разделе сайта разработать страницу «Регистрация пользователя», содержащую форму для ввода (все поля обязательны к заполнению)

Добавлять введенную пользователем информацию в разработанную таблицу базы данных. Предусмотреть проверку на невозможность добавления двух одинаковых логинов.

5. Добавить в меню пользовательского раздела сайта ссылку «Войти» на форму авторизации. Форма должна содержать строку ввода логина, строку ввода пароля, кнопку "Войти", а также ссылку «Регистрация» (на страницу «Регистрация пользователя»).

После успешной авторизации в меню сайта вместо ссылки «Войти» должна отображаться ссылка "Выйти", при нажатии которой происходит выход авторизированного пользователя (завершение текущей сессии).

После успешной авторизации на каждой странице сайта (возможно в области меню) должно отображаться "Пользователь: ФИО" (ФИО – Фами-лия Имя Отчество пользователя).

6. Реализовать отображение результатов тестирования на странице «Тест по дисциплине» только авторизованным пользователям.

1. ХОД РАБОТЫ

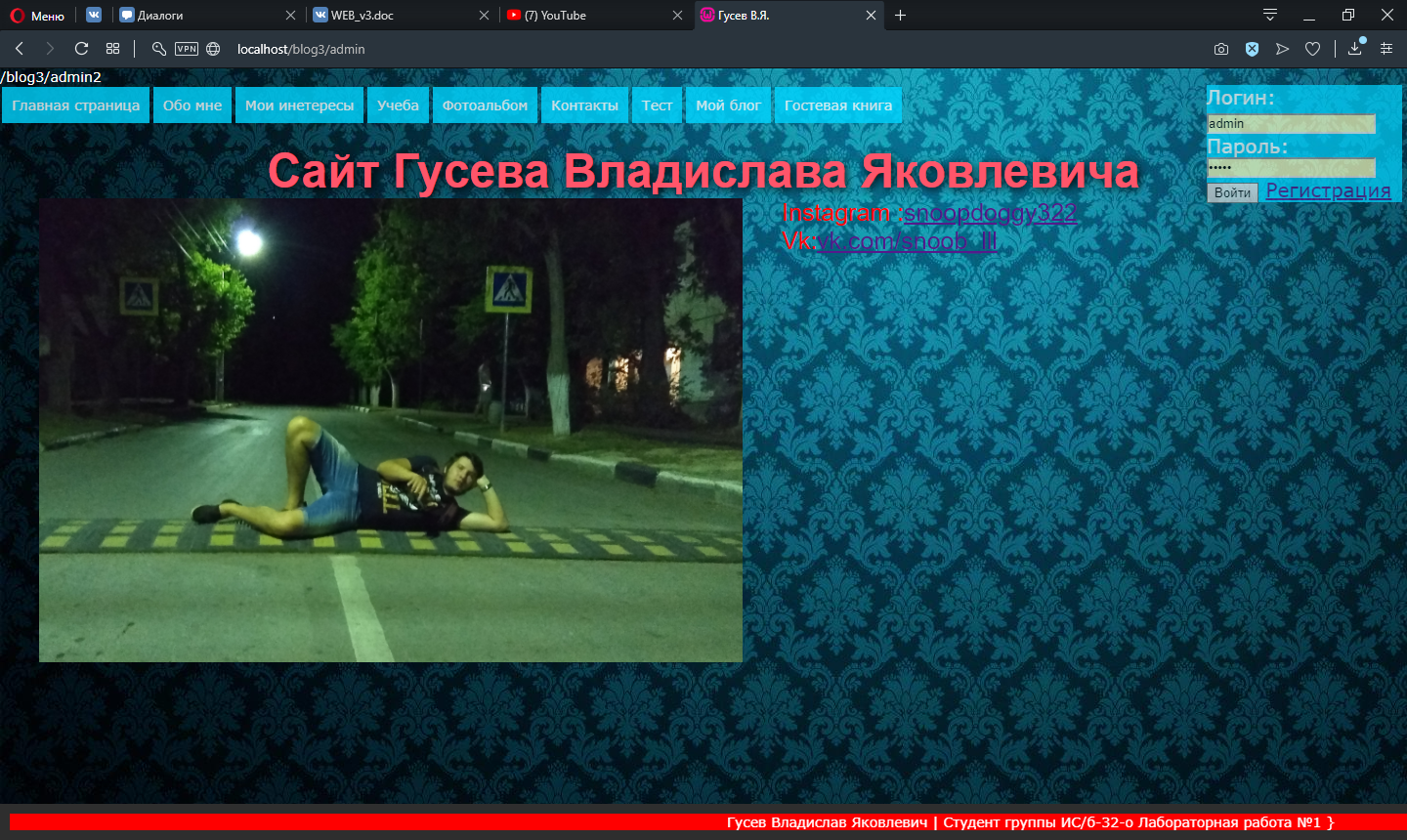


Рисунок 1 – Форма авторизации

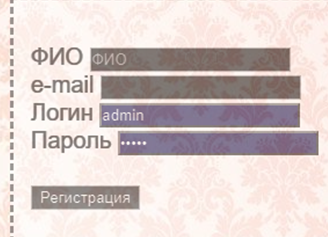


Рисунок 2 – Форма регистрации

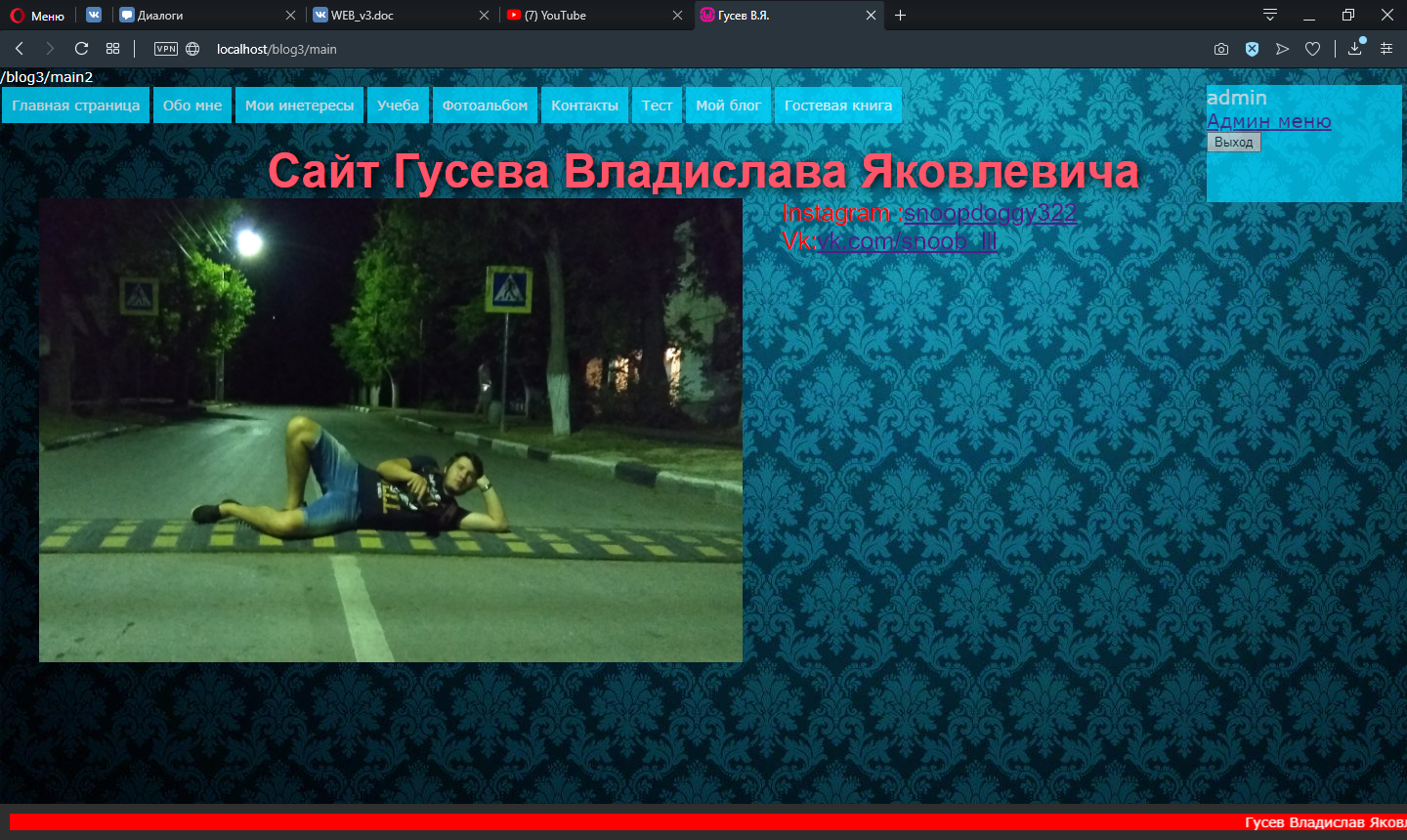


Рисунок 3 –Вид формы авторизованного пользователя

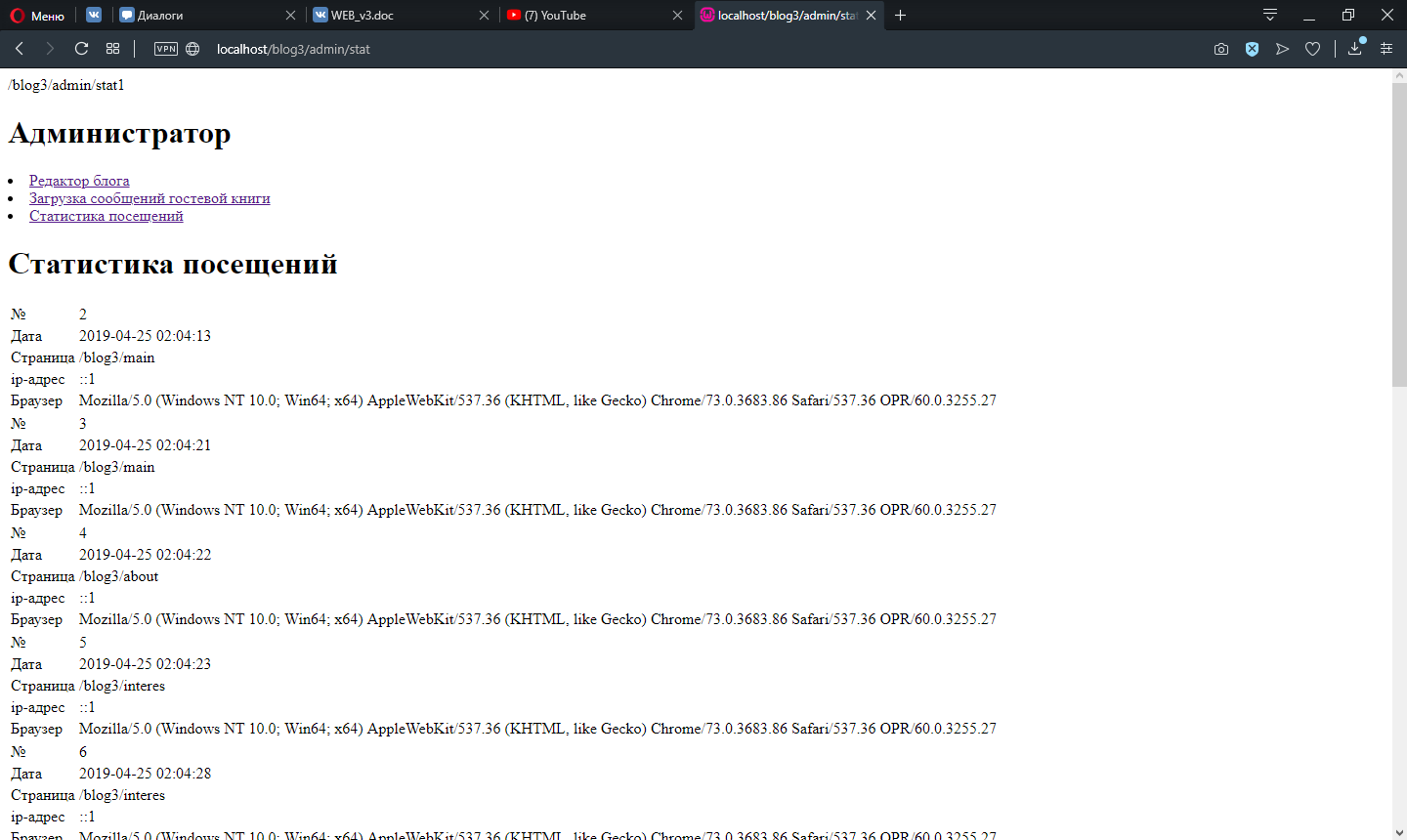


Рисунок 3 – Страница Администратора + Статистика посещений

Класс Login

<?php

/\*\*

\*

\*/

class user

{

function logout ()

{

header('Location: main');

session\_destroy();

}

function enter ()

{

require\_once ('app/models/Model\_users.php');

if ($\_POST['login'] != "" && $\_POST['password'] != "") //если поля заполнены

{

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

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

$q="SELECT \* FROM users WHERE login= '".$login."';";

$rez = UsersModel::find\_q($q); //запрашиваем строку из БД с логином, введённым пользователем

if (count($rez) == 1) //если нашлась одна строка, значит такой юзер существует в БД

{

$row = $rez[0];

//echo md5($password);

if (md5($password) == $row['pass']) //сравниваем хэшированный пароль из БД с хэшированными паролем

{

session\_start();

echo session\_id();

//пишем логин и хэшированный пароль в cookie, также создаём переменную сессии

$\_SESSION['id'] = $row['id']; //записываем в сессию id пользователя

$\_SESSION['name'] = $row['name']; //записываем в сессию id пользователя

if($row['proper']=="admin"){ echo "Вы администратор!";

$\_SESSION['admin'] = $row['proper'];}

else

echo "Авторизация прошла успешно";

header('Location: main');

}

else //если пароли не совпали

{

$error = "Неверный пароль";

}

}

else //если такого пользователя не найдено в БД

{

$error = "Неверный логин и пароль";

}

}

else

{

$error = "Поля не должны быть пустыми!";

}

print($error);

}

function reg ()

{

require\_once( 'app/models/Model\_users.php');

$error = array(); //массив для ошибок

if ($\_POST['login'] != "" && $\_POST['pass'] != "") //если поля заполнены

{

$name = $\_POST['name'];

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

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

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

$params = [0,$name,$email,$login,md5($password),"user"];

if (UsersModel::find\_param("login",$login)!=NULL)

echo "Такой логин уже существует";

else {

UsersModel::save($params);

header('Location: main');

}

}

}

}

if(isset($\_POST['actionFunc'])){

echo "1";

$action\_func = $\_POST['actionFunc'];

if (function\_exists(user::$action\_func())){echo "2s";

$action\_func(); //Здесь вызов функции в нашем случае PlusTime;

}

else{

//Если нет вашего метода то какая то логика

}

}

else{

//Если нет скрытого поля то какая то другая логика

}

Класс Controller\_admin

<?php

class Controller\_admin extends Controller{

function action\_index(){

if($\_SESSION['admin']!=NULL)

$this->view->generate(NULL, 'template\_admin.php');

else

$this->view->generate('main\_view.php', 'template\_view.php');

}

}

Класс – Registration\_view

<body>

<meta charset="utf-8">

<title>Гусев В.Я. Обо мне</title>

<link rel="stylesheet" href="public/css/style1.css">

<body background="1.jpg" >

<header>

<h1>Регистрация</h1>

</header>

<div class ="tex" display="inline-block";>

<form method="POST" enctype="multipart/form-data" name="form\_regisration" style="font-size: 20px"><br>

ФИО <input placeholder="ФИО" maxlength="100" name="name" type= text class="formDesign" required> </input> <br>

<?php

if( isset( $\_POST['accept'] ) ){

if($model && $model->getErrorByField('name')) {

echo "<div class='error'>" . $model->getErrorByField('name')."</div>";

}

}

?>

e-mail <input type="email" maxlength="100" name="email" class="formDesign" required></input><br>

Логин <input type="text" maxlength="100" name="login" class="formDesign" required></input><br>

Пароль <input type="password" maxlength="100" name="pass" class="formDesign" required></input><br>

<?php

?>

<br>

<input type="submit" name="accept" value="Регистрация">

<br>

</div>

</body>

Класс модели пользователя

<?php

include 'app/core/BaseActiveRecord.php';

class UsersModel extends BaseActiveRecord

{

// список полей таблицы

public $id;

public $name;

public $email;

public $login;

public $pass;

public $proper;

public static $table = 'users';

public static function GetParams(){

$st="(id,

name,

email,

login,

pass,

proper)";

return $st;

}

}

// делаем PDO доступным для всех AR классов

BaseActiveRecord::$pdo = $pdo;

// получаем запись из БД с id = 1

//$test = TestModel::find(1);

//var\_dump($test->id);

//var\_dump($test->group);

//var\_dump($test->first);

//var\_dump($test->three);

//var\_dump($test);

ВЫВОДЫ

В ходе выполнения лабораторной работы были изучены возможности авторизации пользователей с использованием механизма сессий PHP, приобрести практические навыки использования переменных, сохраняющих свое значение при переходе от одной страницы сайта к другой.