Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ульяновский государственный технический университет»

Кафедра: «Вычислительная техника»

Дисциплина: «Интернет программирование»

Лабораторная работа №4

«Подключение сайта к БД MySQL»

Выполнил студент

группы ИВТАСбд-12

Рябуша И. А.

Проверил

Игонин А. Г.

Ульяновск, 2022

**Оглавление**

[**Постановка задачи** 3](#_Toc104340576)

[**Особенности реализации** 3](#_Toc104340577)

[**Интерфейс** 6](#_Toc104340578)

[**Вывод** 8](#_Toc104340579)

[**Список литературы** 8](#_Toc104340580)

# **Постановка задачи**

Необходимо реализовать авторизацию на сайте, путем подключения сайта к БД MySQL. Пользователи должны делиться на администраторов и обычных пользователей. Обычные пользователи должны иметь возможность изменять данные своего аккаунта (логин, пароль, имя пользователя, аватар), а также удалять свой аккаунт из БД. Администратор должен иметь возможность редактировать и удалять как аккаунты каждого пользователя, так и свой аккаунт.

# **Особенности реализации**

Для возможности работы с базой данных, используется сторонний класс db.php, который позволяет как записывать данные в БД, так и считывать их.

Для начала необходимо добавить возможность регистрации и авторизации на сайте и параллельную запись пользователей в БД. Для этого, во-первых, проводится валидация введенных пользователем данных:

$temp = "SELECT \* FROM `users` WHERE `login` = '$login'";

if((DB::fetch\_array(DB::query($temp))) == true){

$error = 'Такой пользователь уже зарегистрирован';

}

if (empty($login))

$error .= 'Введите логин!<br>';

if (empty($pass1))

$error .= 'Введите пароль!<br>';

if(strlen($pass1) > 32 || strlen($pass1) < 5){

$error .= 'Длина пароля не должна привышать 32 символов и быть не меньше 5!<br>';

}

if(strlen($login) > 100){

$error .= 'Длина логина не должна привышать 100 символов!<br>';

}

if ($pass1 != $pass2)

$error .= 'Пароли не совпадают!<br>';

Также происходит проверка загрузил ли пользователь аватар и, если да, то путь сохраняется, для последующей передачи в БД, иначе устанавливается дефолтный аватар:

if (($\_FILES['user\_avatar']['name']) != null) {

$uploadTypeFile = "";

if ($\_FILES['user\_avatar']['type'] = 'image/jpeg')

$uploadTypeFile = ".jpg";

$uploadNameFile = md5(time() . $\_FILES['user\_avatar']['name']);

$uploadNameDirection = $\_SERVER['DOCUMENT\_ROOT'] . "/avatars/";

$uploadAvatar = $uploadNameDirection . $uploadNameFile . $uploadTypeFile;

if(!(move\_uploaded\_file($\_FILES['user\_avatar']['tmp\_name'], $uploadAvatar))) {

echo "Возможная атака с помощью файловой загрузки!\n";

echo($\_FILES['user\_avatar']["name"]);

}

$strQueryAvatar = "/avatars/" . $uploadNameFile . $uploadTypeFile;

}else{

$strQueryAvatar = "/avatars/default.jpg";

}

После валидации происходит запись в БД:

if (empty($error)){

$query = "INSERT INTO `users` (`login`, `username`, `password`, `avatar\_name`) VALUES ('$login', '$username', MD5('$pass1'), '$strQueryAvatar')";

DB::query($query);

$\_SESSION['auth'] = true;

$\_SESSION['name'] = $username;

}

При авторизации, в БД отправляется запрос о данных пользователя:

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$llogin = htmlspecialchars($\_POST['user\_login']);

$pass = htmlspecialchars($\_POST['user\_password']);

$error = '';

if (empty($llogin) || empty($pass)) {

$error = 'Ошибка авторизации, поля не должны быть пустыми!';

}

if (empty($error)) {

$query = "SELECT \* FROM `users` WHERE `login` = '$llogin' and `password` = MD5('$pass')";

$res = DB::query($query);

if(($item = DB::fetch\_array($res)) != false) {

$\_SESSION['auth'] = true;

$\_SESSION['name'] = $item['username'];

setcookie('id', $item['id'], 0, "/");

header('location: /');

}else{

$error = 'Неверное имя пользователя или пароль';

}

}

include\_once "../reg\_log.php";

}

Для изменения данных, как со стороны пользователя, так и администратора, используется SQL запрос UPDATE, который выполняется после валидации, идентичной той что была при регистрации.

$query = "UPDATE `users`

SET `login`= '".$\_POST['login']."', `username` = '".$\_POST['username']."'"

. $strQueryPass . $strQueryAvatar . "

WHERE id=".$\_POST['id'];

$res = DB::query($query);

При удалении происходит запрос о роли пользователя. Если пользователь админнистратор, тогда SQL запросом DELETE удаляется выбранный пользователь, а администратор возвращается на панель со списком пользователей.

$isAdmin = DB::fetch\_array(DB::query("SELECT \* FROM `users` WHERE id=".$\_COOKIE['id']));

$query = "SELECT \* FROM `users` WHERE id=".$\_GET['id'];

$res = DB::query($query);

if( ($item = DB::fetch\_array($res)) != false) {

if($isAdmin['type'] == '0') {

$query = "DELETE FROM `users`

WHERE id=" . $item['id'];

$res = DB::query($query);

if($isAdmin['id'] != $item['id']){

header("location: /admin/userlist.php");

}else{

include\_once "../exit.php";

}

Если же пользователь удаляет свой аккаунт, либо администратор свой, то помимо запроса исполняется файл exit.php очищающий данные сессии.

}else{

$query = "DELETE FROM `users`

WHERE id=" . $item['id'];

$res = DB::query($query);

include\_once "../exit.php";

}

}

Exit.php:

session\_start();

unset($\_SESSION['auth']);

unset($\_SESSION['name']);

header('location: /');

# **Интерфейс**

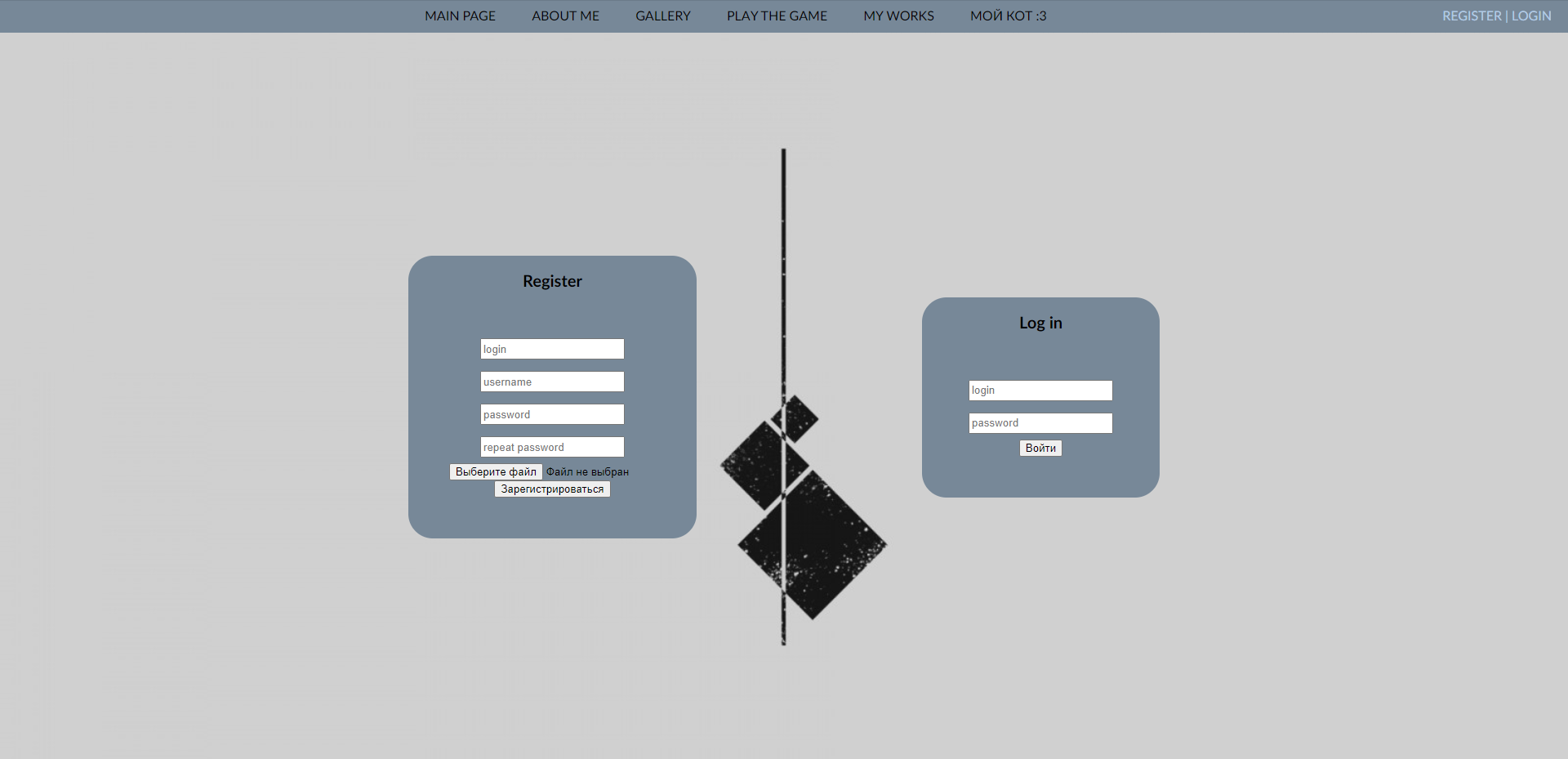


Рис. 1 – Страница регистрации и авторизации

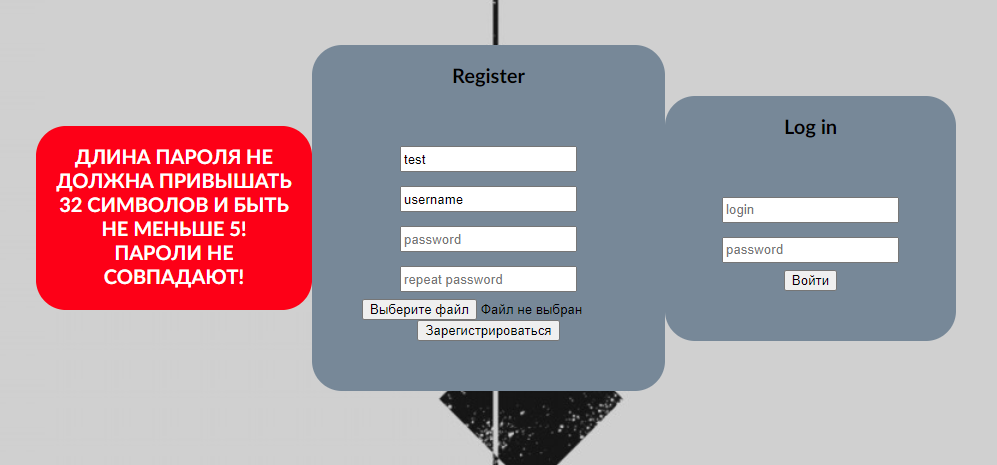


Рис. 2 – Ошибка при некорректном вводе данных

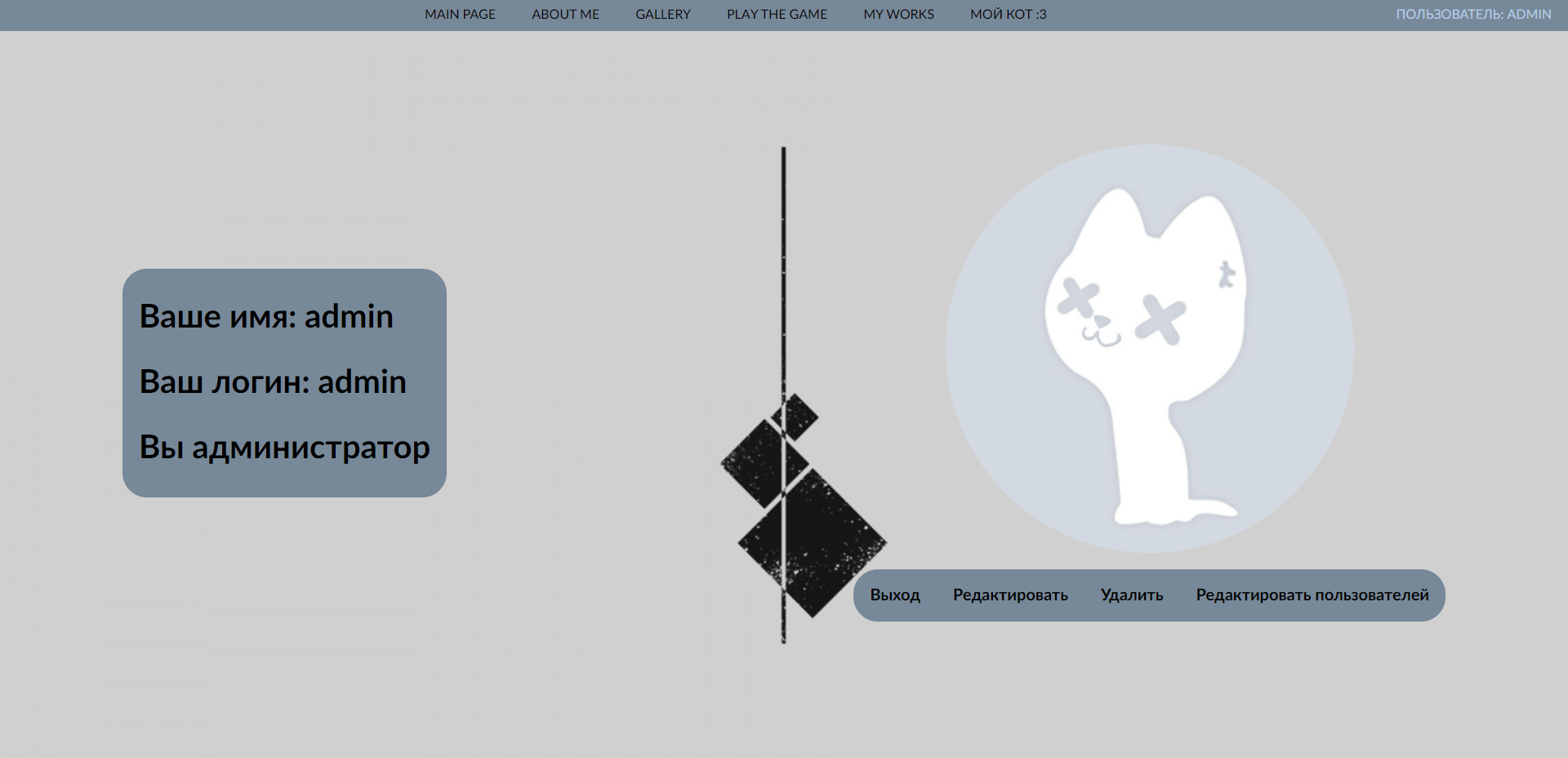


Рис. 3 – Страница пользователя в роли администратор

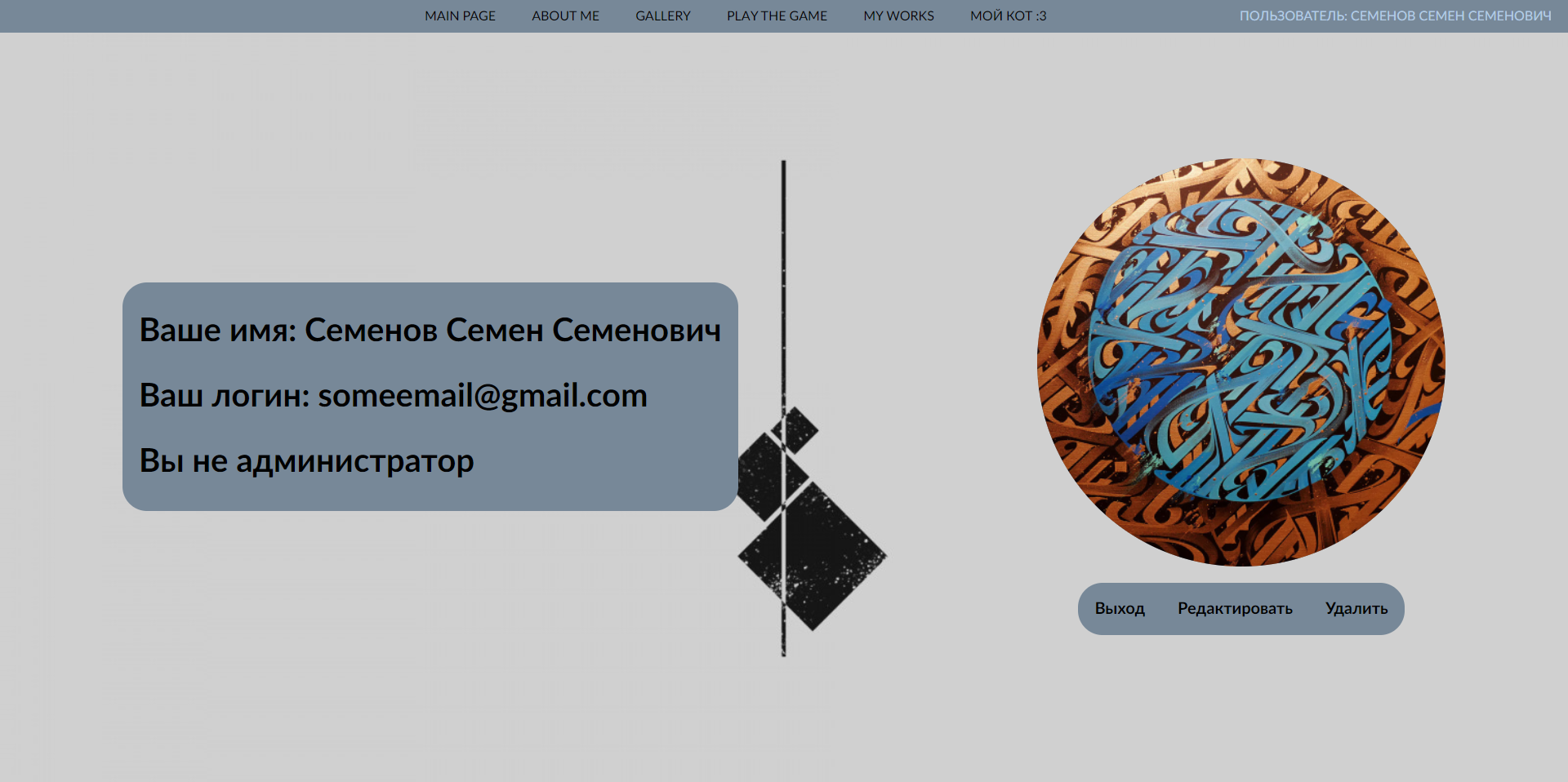


Рис. 4 – Страница стандартного пользователя



Рис. 5 – Список пользователей. доступный администратору

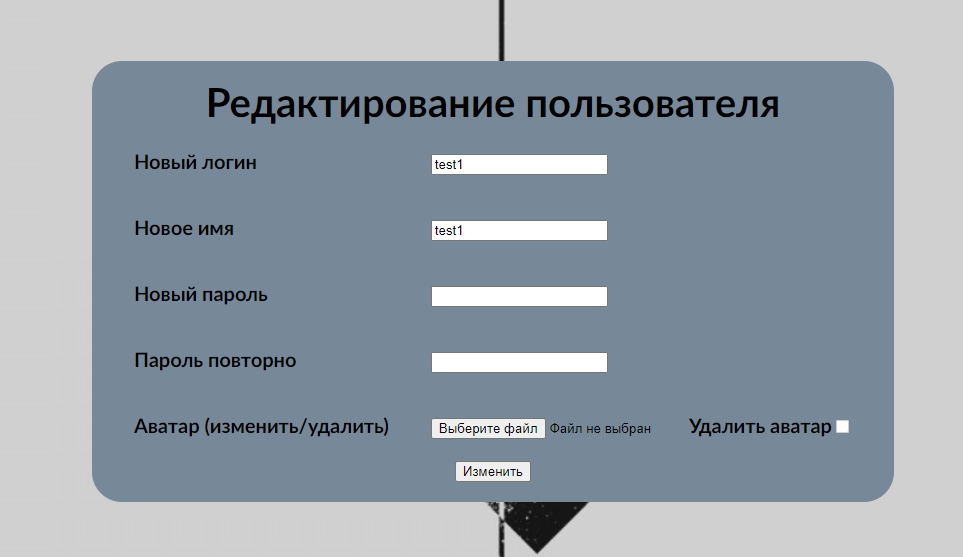


Рис. 6 – Панель редактирования данных

# **Вывод**

В результате проделанной работы, к сайту была подключена база данных MySQL. С помощью языка PHP, были реализованы роли пользователей и соответствующие привилегии, такие как редактирование данных пользователей, удаление пользователей, редактирование/удаление/добавление аватарки пользователя.

# **Список литературы**

1. Справочник по PHP. [Электронный ресурс] URL: https://www.php.net/manual/ru/langref.php
2. Справочник по MySQL. [Электронный ресурс] URL: https://metanit.com/sql/mysql/