

Аутентификация и авторизация с COOKIE

- Введение
- Практика аутентификации
- Отмена аутентификации

Введение

Аутентификация, построенная на практике применения файлов **cookie** – это скорее для тех идеалистов, которые считают, что:

- Мир разумен и справедлив!
- Жизнь прекрасна и удивительна!
- Люди милы и доброжелательны!

Для тех, кто усомнился в заявленных аксиомах есть аутентификация и авторизация, но построенные на **сессиях**. Именно с использованием сессий создаются реальные приложения.

И тем не менее. Алгоритм аутентификации и тонкости безопасной работы рассмотрим именно здесь.

К тому же познакомимся с важными функциями обработки паролей, сохраненных в базе данных в виде хешей.

В файле раздаточного материала **dump/db_auth.sql** предлагается **дамп базы данных** для выполнения демонстрационных примеров. Дамп состоит из трех таблиц, каждая из таблиц предназначена для отработки следующих навыков:

- **user_open** – аутентификация с открытым паролем;
- **user_hash** – аутентификация с хешем пароля от функции **password_hash()**;
- **user_crypt** – аутентификация с хешем пароля от функции **crypt()**.

Дамп представлен с пользователями по умолчанию. Но прежде, чем приступить к непосредственной теме, создадим сценарий, регистрирующий **дополнительного пользователя** для каждой из таблиц предложенной базы данных.

Примечание. Практика регистрация пользователя и создания хеша пароля рассмотрена в теме **Регистрация и COOKIE**.

Форма регистрации.

example_1. index.html

```
<form action="reg.php" method="post">

    Логин: <input type="text" name="login" value=""><p>

    Пароль: <input type="text" name="pwd" value=""><p>

    Открытый пароль: <input type="radio" name="hash" value="none"
checked><p>

    Хеш пароля функцией crypt (+ salt): <input type="radio"
name="hash" value="crypt"><p>

    Хеш пароля функцией password_hash: <input type="radio"
name="hash" value="pwd_hash"><p>

    Роль :

<select name="role">

    <option value="user">Пользователь</option>

    <option value="moderator">Модератор</option>

    <option value="admin">Администратор</option>

</select><p>
```

```
<button>Регистрация</button>

</form>
```

Обратите внимание, в обработчике формы происходит **выбор таблицы** для вставки нового пользователя.

example_1. reg.php

```
<?php

// подключение к базе данных

$mysqli = new mysqli ('localhost', 'root', '', 'db_auth');

/*

    обратите внимание, запись может быть выполнена

    в таблицы БД:

    -- user_open

    -- user_crypt

    -- user_hash

*/

switch ($_POST['hash']) {

    case 'pwd_hash':

        define ('TBL_NAME', 'user_hash');

        break;

    case 'crypt':

        define ('TBL_NAME', 'user_crypt');

        break;

    case 'none':

        define ('TBL_NAME', 'user_open');

}

// подготовка / привязка параметров / выполнение запроса
```

```

// определяем требуемое хеширование пароля

if($_POST['hash'] == 'pwd_hash'){

    $insert = 'INSERT INTO `'. TBL_NAME . '` SET `login` =
    ?, `pwd` = ?, `role` = ?';

    $stmt = $mysqli->prepare($insert);

    $stmt->bind_param('sss', $_POST['login'], $pwd,
    $_POST['role']);

    $pwd = password_hash($_POST['pwd'], PASSWORD_DEFAULT);

}

elseif ($_POST['hash'] == 'crypt') {

    $insert = 'INSERT INTO `'. TBL_NAME . '` SET `login` =
    ?, `pwd` = ?, `salt` = ?, `role` = ?';

    $stmt = $mysqli->prepare($insert);

    $stmt->bind_param('ssss', $_POST['login'], $pwd, $salt,
    $_POST['role']);

    $chars = '0123456789abcdefghijklmnopqrstuvwxyz';

    $salt = substr(str_shuffle($chars), 0, 10);

    $pwd = crypt($_POST['pwd'], $salt);

} else {

    $insert = 'INSERT INTO `'. TBL_NAME . '` SET `login` =
    ?, `pwd` = ?, `role` = ?';

    $stmt = $mysqli->prepare($insert);

    $stmt->bind_param('sss', $_POST['login'], $_POST['pwd'],
    $_POST['role']);

}

// если запрос выполнен успешно

if ($stmt->execute()) {

    // получаем id вставленной записи

```

```

$id = $stmt->insert_id;

// подготовка / привязка параметров / выполнение запроса

$stmt = $mysqli->prepare('SELECT * FROM ` ` . TBL_NAME .
` ` WHERE `id_user` = ?');

$stmt->bind_param('d', $id);

$stmt->execute();

// получение результата запроса

$result = $stmt->get_result();

$row = $result->fetch_assoc();

echo <<<HERE

    Пользователь <br>

    <ul>

        <li>Идентификатор: {$row['id_user']} </li>

        <li>Логин: {$row['login']} </li>

        <li>Пароль: {$row['pwd']} </li>

    </ul>

    <b>успешно зарегистрирован</b><p>

    <a href="/">Назад к форме регистрации</a>

    HERE;

// если ошибка вставки

} else {

    echo '<h3>Что-то пошло не так :(</h3>';

    echo '<a href="/">Попробовать еще раз</a>';

};

?>

```

Теперь выполним практику входа пользователя в систему и предоставления ему определенных прав. Представленные примеры –

отправная точка для **самостоятельных экспериментов**. Требование одно – мы используем файлы **cookie**.

Практика аутентификации

Каждый демонстрационный пример будет содержать три директории для тестирования прав пользователя – **авторизации**.

1. **admin** – директория тестирования пользователей с правами администратора (**admin**);
2. **moderator** – директория тестирования пользователей с правами модератора (**moderator**) или администратора (**admin**);
3. **user** – директория тестирования неаутентифицированных пользователей и пользователей с правами **user**.

| Имя | Состояние | Дата изменения | Тип | Размер |
|------------|-----------|------------------|--------------------|--------|
| admin | | 28.12.2023 10:50 | Папка с файлами | |
| moderator | | 28.12.2023 10:50 | Папка с файлами | |
| user | | 28.12.2023 10:50 | Папка с файлами | |
| auth.php | | 29.12.2023 12:32 | Исходный файл Р... | 3 КБ |
| index.html | | 26.12.2023 16:48 | Chrome HTML Do... | 1 КБ |

Аутентификация с открытым паролем

Начнем тестирование аутентификации на таблице **user_open**. Фрагмент структуры и пользователи таблицы представлены ниже.

| user_open | | | |
|-----------|----------------|--------------|-----------|
| № | Логин | Пароль | Роль |
| 1 | loginUser | pwdUser | user |
| 2 | loginModerator | pwdModerator | moderator |
| 3 | loginAdmin | pwdAdmin | admin |

Все табличные данные можете поменять под ваши предпочтения.

Форма для входа в систему представлена в файле **index.html**.

example_2. index.html

```
<form action="auth.php" method="post">

    Логин: <input type="text" name="login" value=""><p>

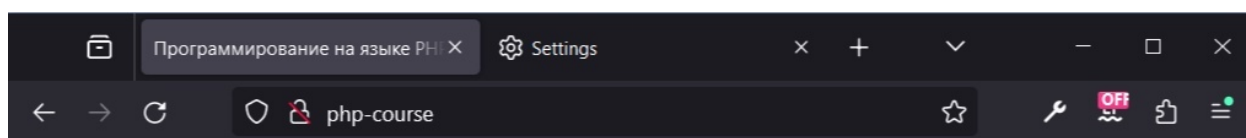
    Пароль: <input type="text" name="pwd" value=""><p>

    <button>Войти</button>

</form>

<p><a href='auth.php'>Пропустить аутентификацию</a></p>
```

Внешний вид формы с аутентификационными данными пользователя **user** представлен на скриншоте.



Аутентификация и авторизация с COOKIE

Логин:

Пароль:

[Пропустить аутентификацию](#)

Обработчиком формы и, соответственно, сценарием, выполняющим аутентификацию пользователя, является файл **auth.php**.

В демо-примере **example_2** рассмотрим самый простой вариант – передачу данных пользователя в виде JSON-представления массива в заголовках **cookie**. И если принцип от простого к сложному все еще актуален, далее будем усложнять...

example_2. auth.php

```
<?php

// если есть POST-данные формы

if (!empty($_POST)) {

    // подключение к базе данных

    $mysqli = new mysqli ('localhost', 'root', '',
        'db_auth');

    // создание, выполнение подготовленного запроса

    $stmt = $mysqli->prepare('SELECT * FROM `user_open` WHERE
        `login` = ?');

    $stmt->bind_param('s', $_POST["login"]);

    $stmt->execute();

    // получение результата

    $result = $stmt->get_result();

    // если запрос вернул запись

    if ($result->num_rows) {

        $row = $result->fetch_assoc();

        // если пароли совпадают

        if ($_POST['pwd'] == $row['pwd']) {

            $msg = "<h2>Поздравляю! Вы успешно
                аутентифицированы!</h2>";

            $msg .= "<p>Вход выполнил пользователь:
                {$_POST["login"]}</p>";

            // подготовка строки JSON для сохранения в cookie

            $user = json_encode(array(

                'login' => $row['login'],

                'pwd' => $row['pwd'],

                'role' => $row['role']
```

```

    ));

    // установка cookie

    setcookie('user', $user, 0, '/');

} else {

    $msg = "<h3>Неправильно введен пароль</h3>";

    $msg .= "<p><a href='/'>Попробовать еще
    раз</a></p>";

}

}

else {

    $msg = "<h3>Пользователь с таким логином не
    найден</h3>";

    $msg .= "<p><a href='/'>Попробовать еще раз</a></p>";

}

}

elseif (isset($_COOKIE['user'])) {

    // декодируем JSON

    $user = json_decode($_COOKIE['user'], true);

    $msg = "<p>Вы продолжаете работать как:
    {$user['login']}</p>";

}

else

    $msg = "<p>Выполнить <a href='/'>вход</a> вы можете в
    любое удобное для вас время</p>";

?>

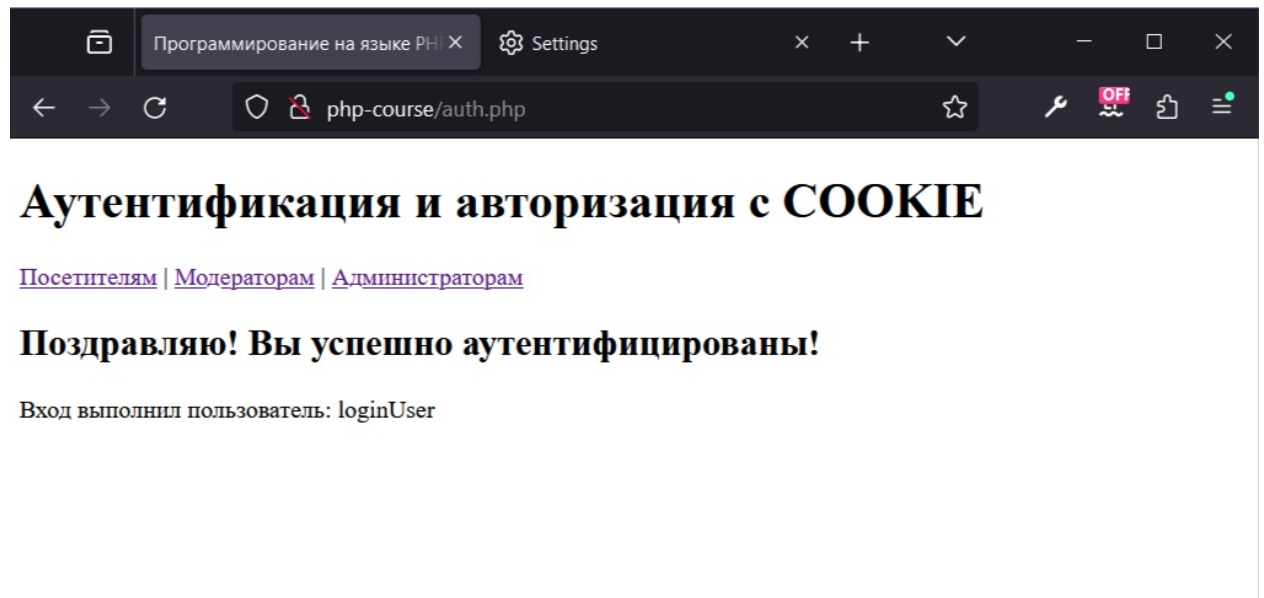
<!-- ... -->

<!-- меню сайта -->

<p>

```

```
<a href='/user/'>Посетителям</a> |  
  
<a href='/moderator/'>Модераторам</a> |  
  
<a href='/admin/'>Администраторам</a>  
  
</p>  
  
<?php echo $msg; ?>
```



Аутентификация выполнена, данные пользователя сохранены в **cookie**.
Теперь на каждой из страниц сайта осталось их **извлечь**,
декодировать и **проанализировать** для предоставления
соответствующих привилегий текущему пользователю.

В директории **user** хранится файл **index.php**, отвечающий за вывод
страницы **Посетителям**.

example_2. user/index.php

```
<h2>Добро пожаловать всем! ;) </h2>  
  
<?php  
  
    // если аутентификация состоялась  
  
    if (isset($_COOKIE['user'])) {
```

```
// декодируем JSON

$user = json_decode($_COOKIE['user'], true);

// проверка роли для доступа к странице

if ($user['role'] == 'user') {

    echo '<h2>Для зарегистрированных пользователей
    презент! ;)</h2>';

}

}
```

Все работает, но есть одна маленькая проблема (или не очень маленькая). Все данные пользователя видны как на витрине магазина.

Аутентификация и авторизация с COOKIE

[Посетителям](#) | [Модераторам](#) | [Администраторам](#)

Добро пожаловать всем! ;)

Для зарегистрированных пользователей презент! ;)

Inspector Network

2 requests 2.23 kB / 1.08 kB transferred Finish: 103 ms

Headers Cookies Request Response Timings

Filter Cookies

Request Cookies

user: '("login":"loginUser","pwd":"pwdUser","role":"user")'

Это и есть тот недостаток, который делает передачу аутентификационных данных пользователя через файлы **cookie** возможной только в эфемерном мире добра и справедливости...

Примечание. Вопросы безопасной аутентификации будем рассматривать в следующих темах.

Директория **moderator** отвечает за вывод страницы **Модераторам**.

example_2. moderator/index.php

```
<?php

// если аутентификация состоялась

if (isset($_COOKIE['user'])) {

    // декодируем JSON

    $user = json_decode($_COOKIE['user'], true);

    // проверка роли для доступа к странице

    if (

        $user['role'] == 'admin' ||

        $user['role'] == 'moderator'

    ) {

        echo '<h2>Добро пожаловать! ;)</h2>';

    } else {

        echo '<h2>Контент только для пользователей с правами администратора или модератора...</h2>';

        echo '<p><a href="/">Выполнить вход</a></p>';

    }

}

else {

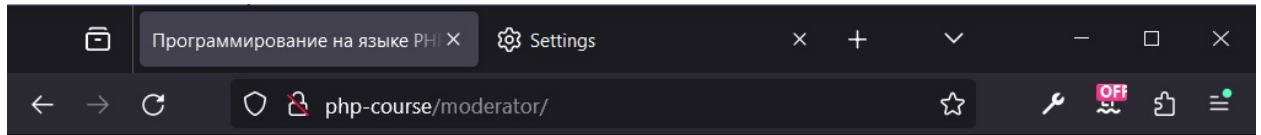
    echo '<h2>Для просмотра контента необходима аутентификация!</h2>';

    echo '<p><a href="/">Выполнить вход</a></p>';

}
```

?>

Страница **Модераторам** для пользователя с правами **user** представлена на скриншоте.



Аутентификация и авторизация с COOKIE

[Посетителям](#) | [Модераторам](#) | [Администраторам](#)

Контент только для пользователей с правами администратора или модератора...

[Выполнить вход](#)

Директория **admin** хранит файл **index.php**, отвечающий за вывод страницы **Администраторам**.

example_2. admin/index.php

```
<?php

// если аутентификация состоялась

if (isset($_COOKIE['user'])) {

    // декодируем JSON

    $user = json_decode($_COOKIE['user'], true);

    // проверка роли для доступа к странице

    if ($user['role'] == 'admin') {

        echo '<h2>Добро пожаловать! ;)</h2>';

    } else {

        echo '<h2>Контент только для администраторов!</h2>';

    }

}
```

```

        echo '<p><a href="/">Выполнить вход</a></p>';

    }

    } else {

        echo '<h2>Для просмотра контента необходима
аутентификация!</h2>';

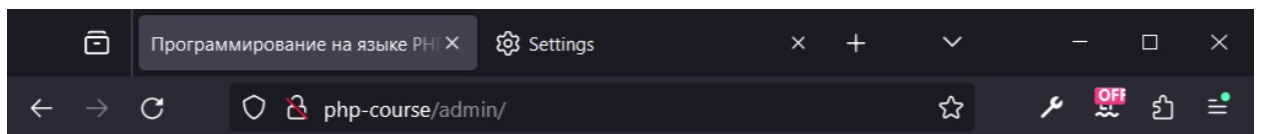
        echo '<p><a href="/">Выполнить вход</a></p>';

    }

?>

```

Страница **Администраторам** для пользователя с правами **user** представлена на скриншоте.



Аутентификация и авторизация с COOKIE

[Посетителям](#) | [Модераторам](#) | [Администраторам](#)

Контент только для администраторов!

[Выполнить вход](#)

Следующие эксперименты проведем на **хешированном пароле**.

Аутентификация с хешем пароля от функции password_hash()

Тестирование аутентификации с паролем в виде хеша начнем с таблицы **user_hash**. Ниже представлены фрагмент структуры и пользователи таблицы.

| № | Логин | Пароль | Роль | Примечание |
|---|----------------|--------------|-----------|------------|
| 1 | loginUser | pwdUser | user | хеш |
| 2 | loginModerator | pwdModerator | moderator | хеш |
| 3 | loginAdmin | pwdAdmin | admin | хеш |

Функция **password_hash()** не требует хранения соли для хеширования пароля. Сравнение введенных данных и данных, хранящихся в базе выполняется с помощью функции **password_verify()**.

password_verify

password_verify() — проверяет, соответствует ли пароль хешу.

Описание

```
password_verify($password, $hash);
```

Проверяет, соответствует ли пароль хешу. Функция **password_verify()** совместима с **crypt()**. Следовательно, хеши паролей, созданные **crypt()**, могут быть использованы в **password_verify()**.

Примечание. Это означает, что хеш, созданный функцией **crypt()** может быть также проверен функцией **password_verify()**. Разберем далее.

Обратите внимание, что **password_hash()** возвращает **алгоритм**, **стоимость** и **соль** как **части хеша**. Таким образом, вся необходимая для проверки информация включена в сам хеш. Это позволяет производить проверку без необходимости хранить все вспомогательные данные отдельно.

Примечание. **Password_hash()** также как и **crypt()** использует соль.

Отличается способ хранения соли.

Эта функция безопасна для атак по времени.

Параметры

- **password** – пользовательский пароль.
- **hash** – хеш, созданный функцией **password_hash()**.

Возвращаемые значения

Возвращает **true** или **false**, в зависимости от результатов проверки.

example_3. index.html

```
<form action="auth.php" method="post">

    Логин: <input type="text" name="login" value=""><p>

    Пароль: <input type="text" name="pwd" value=""><p>

    <button>Войти</button>

</form>

<p><a href='auth.php'>Пропустить аутентификацию</a></p>
```

example_3. auth.php

```
<?php

    // если есть POST-данные формы

    if (!empty($_POST)) {

        // подключение к базе данных

        $mysqli = new mysqli ('localhost', 'root', '',
            'db_auth');

        // создание, выполнение подготовленного запроса

        $stmt = $mysqli->prepare('SELECT * FROM `user_hash` WHERE
            `login` = ?');
```

```
$stmt->bind_param('s', $_POST["login"]);

$stmt->execute();

// получение результата

$result = $stmt->get_result();

// если запрос вернул запись

if ($result->num_rows) {

    $row = $result->fetch_assoc();

    // если пароли совпадают

    if (password_verify($_POST['pwd'], $row['pwd'])) {

        $msg = "<h2>Поздравляю! Вы успешно  
аутентифицированы!</h2>";

        $msg .= "<p>Вход выполнил пользователь:  
{$_POST["login"]}</p>";

        // установка cookie

        setcookie('user[login]', $_POST["login"], 0, '/');

        setcookie('user[pwd]', $_POST['pwd'], 0, '/');

    } else {

        $msg = "<h3>Неправильно введен пароль</h3>";

        $msg .= "<p><a href='/'>Попробовать еще  
раз</a></p>";

    }

}

else {

    $msg = "<h3>Пользователь с таким логином не  
найден</h3>";

    $msg .= "<p><a href='/'>Попробовать еще раз</a></p>";

}
```

```

    } elseif (isset($_COOKIE['user']))

        $msg = "<p>Вы продолжаете работать как:
        {$_COOKIE['user']['login']}

```

В каждом из демонстрационных примеров буду приводить полный код **PHP-сценариев**. Внимательно анализируйте, коды примеров не будут повторяться один в один.

example_3. user/index.php

```

<?php

    // если аутентификация состоялась

    if (isset($_COOKIE['user'])) {

        // подключение к базе данных

        $mysqli = new mysqli ('localhost', 'root', '',
        'db_auth');

        // создание, выполнение подготовленного запроса

        $stmt = $mysqli->prepare('SELECT * FROM `user_hash` WHERE

```

```

        `login` = ?');

$stmt->bind_param('s', $_COOKIE['user']['login']);

$stmt->execute();

// получение результата

$res = $stmt->get_result();

$row = $res->fetch_assoc();

// проверка совпадения паролей

// проверка роли для доступа к странице

if (

    password_verify($_COOKIE['user']['pwd'], $row['pwd'])
    &&

    ($row['role'] == 'user' ||

    $row['role'] == 'admin' ||

    $row['role'] == 'moderator')

) {

    echo '<h2>Для зарегистрированных пользователей
    презент! ;)</h2>';

}

}

?>

```

example_3. moderator/index.php

```

<?php

// если аутентификация состоялась

if (isset($_COOKIE['user'])) {

    // подключение к базе данных

    $mysqli = new mysqli ('localhost', 'root', '',

```

```

'db_auth');

// создание, выполнение подготовленного запроса

$stmt = $mysqli->prepare('SELECT * FROM `user_hash` WHERE
`login` = ?');

$stmt->bind_param('s', $_COOKIE['user']['login']);

$stmt->execute();

// получение результата

$res = $stmt->get_result();

$row = $res->fetch_assoc();

// проверка совпадения паролей

// проверка роли для доступа к странице

if (

    password_verify($_COOKIE['user']['pwd'], $row['pwd'])
    &&

    ($row['role'] == 'admin' ||

    $row['role'] == 'moderator')

) {

    echo '<h2>Добро пожаловать! ;)</h2>';

} else {

    echo '<h2>Контент только для пользователей с правами
администратора или модератора...</h2>';

    echo '<p><a href="/">Выполнить вход</a></p>';

}

}

else {

    echo '<h2>Для просмотра контента необходима
аутентификация!</h2>';

    echo '<p><a href="/">Выполнить вход</a></p>';

```

```
}
```

```
?>
```

example_3. admin/index.php

```
<?php

// если аутентификация состоялась

if (isset($_COOKIE['user'])) {

    // подключение к базе данных

    $mysqli = new mysqli ('localhost', 'root', '',
        'db_auth');

    // создание, выполнение подготовленного запроса

    $stmt = $mysqli->prepare('SELECT * FROM `user_hash` WHERE
        `login` = ?');

    $stmt->bind_param('s', $_COOKIE['user']['login']);

    $stmt->execute();

    // получение результата

    $res = $stmt->get_result();

    $row = $res->fetch_assoc();

    // проверка совпадения паролей

    // проверка роли для доступа к странице

    if (

        password_verify($_COOKIE['user']['pwd'], $row['pwd'])
        &&

        $row['role'] == 'admin'

    ) {

        echo '<h2>Добро пожаловать! ;)</h2>';

    } else {
```

```

        echo '<h2>Контент только для администраторов!</h2>';

        echo '<p><a href="/">Выполнить вход</a></p>';

    }

} else {

    echo '<h2>Для просмотра контента необходима
аутентификация!</h2>';

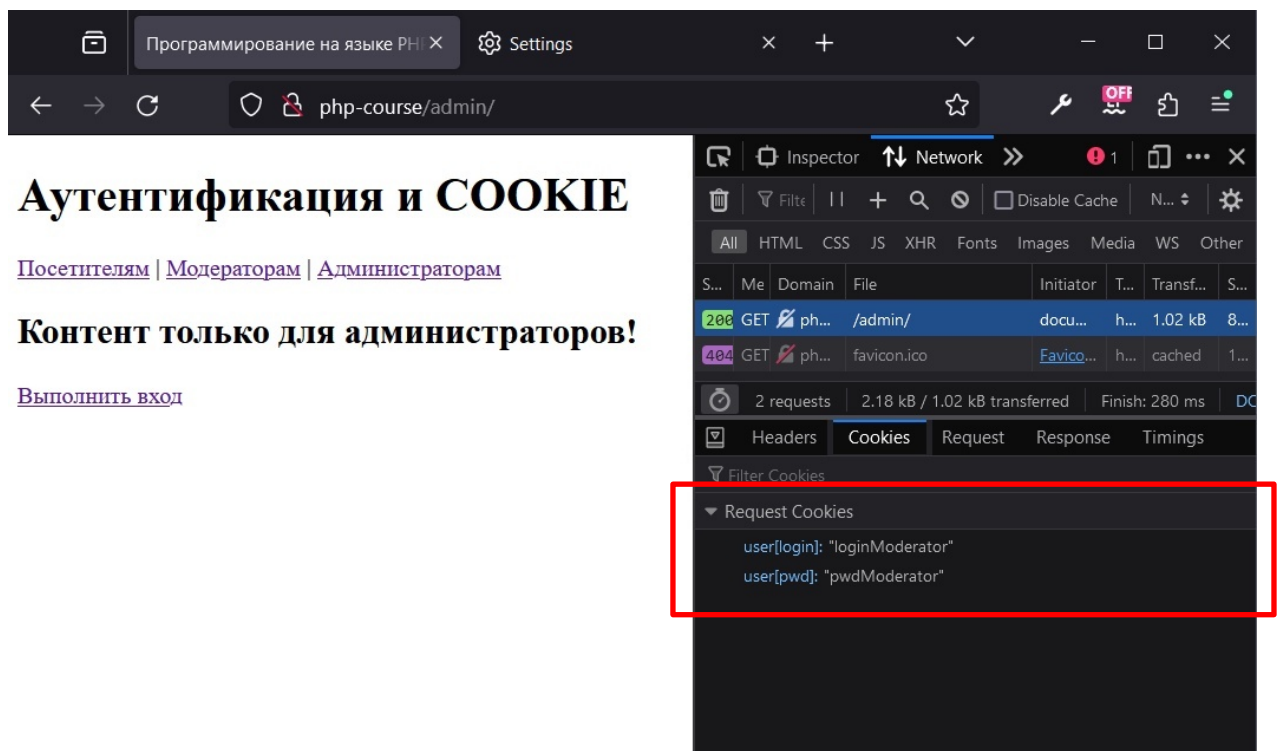
    echo '<p><a href="/">Выполнить вход</a></p>';

}

?>

```

И опять данные модератора как на ладони.



Хэш защищает идентификационные данные пользователя **на уровне базы данных**, но не защищает **на уровне приложения**.

Аутентификация с хешем пароля от функции crypt()

Следующие примеры аутентификации с хешем выполним на таблице **user_crypt**. Фрагмент структура и пользователи таблицы представлены ниже.

| user_crypt | | | | |
|------------|----------------|--------------|-----------|------------|
| № | Логин | Пароль | Роль | Примечание |
| 1 | loginUser | pwdUser | user | хеш |
| 2 | loginModerator | pwdModerator | moderator | хеш |
| 3 | loginAdmin | pwdAdmin | admin | хеш |

example_4. index.html

```
<form action="auth.php" method="post">

    Логин: <input type="text" name="login" value=""><p>

    Пароль: <input type="text" name="pwd" value=""><p>

    <button>Войти</button>

</form>

<p><a href='auth.php'>Пропустить аутентификацию</a></p>
```

example_4. auth.php

```
<?php

    // если есть POST-данные формы

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

        // подключение к базе данных

        $mysqli = new mysqli ('localhost', 'root', '',
            'db_auth');

        // создание, выполнение подготовленного запроса

        $stmt = $mysqli->prepare('SELECT * FROM `user_crypt`
            WHERE `login` = ?');
```

```
$stmt->bind_param('s', $_POST["login"]);

$stmt->execute();

// получение результата

$result = $stmt->get_result();

// если запрос вернул запись

if ($result->num_rows) {

    $row = $result->fetch_assoc();

    // получаем хеш от пароля

    $hash = crypt($_POST['pwd'], $row['salt']);

    // если пароли совпадают

    if ($hash == $row['pwd']) {

        $msg = "<h2>Поздравляю! Вы успешно  
аутентифицированы!</h2>";

        $msg .= "<p>Вход выполнил пользователь:  
{$_POST["login"]}</p>";

        // установка cookie

        setcookie('user[login]', $_POST["login"], 0,
            '/');

        setcookie('user[pwd]', $_POST['pwd'], 0, '/');

    } else {

        $msg = "<h3>Неправильно введен пароль</h3>";

        $msg .= "<p><a href='/'>Попробовать еще  
раз</a></p>";

    }

}

else {

    $msg = "<h3>Пользователь с таким логином не  
найден</h3>";
```

```

        $msg .= "<p><a href='/'>Попробовать еще раз</a></p>";
    }

}

elseif (isset($_COOKIE['user']))

    $msg = "<p>Вы продолжаете работать как:
    {$_COOKIE['user']['login']}</p>";

else

    $msg = "<p>Выполнить <a href='/'>вход</a> вы можете в
    любое удобное для вас время</p>";

?>

<!-- ... -->

<!-- меню сайта -->

<p>

    <a href='/user/'>Посетителям</a> |

    <a href='/moderator/'>Модераторам</a> |

    <a href='/admin/'>Администраторам</a>

</p>

<?php echo $msg; ?>

```

example_4. user/index.php

```

<?php

    // если аутентификация состоялась

    if (isset($_COOKIE['user'])) {

        // подключение к базе данных

        $mysqli = new mysqli ('localhost', 'root', '',
        'db_auth');

        // создание, выполнение подготовленного запроса

        $stmt = $mysqli->prepare('SELECT * FROM `user_crypt`

```

```

WHERE `login` = ?');

$stmt->bind_param('s', $_COOKIE['user']['login']);

$stmt->execute();

// получение результата

$res = $stmt->get_result();

$row = $res->fetch_assoc();

// получаем хеш пароля

$hash = crypt($_COOKIE['user']['pwd'], $row['salt']);

// проверка совпадения паролей

// проверка роли для доступа к странице

if (

    $hash == $row['pwd'] &&

    ($row['role'] == 'user' ||

    $row['role'] == 'admin' ||

    $row['role'] == 'moderator')

) {

    echo '<h2>Для зарегистрированных пользователей

    презент! ;</h2>';

}

}

```

?>

example_4. moderator/index.php

```

<?php

// если аутентификация состоялась

if (isset($_COOKIE['user'])) {

    // подключение к базе данных

```

```

$mysqli = new mysqli ('localhost', 'root', '',
'db_auth');

// создание, выполнение подготовленного запроса

$stmt = $mysqli->prepare('SELECT * FROM `user_crypt`
WHERE `login` = ?');

$stmt->bind_param('s', $_COOKIE['user']['login']);

$stmt->execute();

// получение результата

$res = $stmt->get_result();

$row = $res->fetch_assoc();

// получаем хеш пароля

$hash = crypt($_COOKIE['user']['pwd'], $row['salt']);

// проверка совпадения паролей

// проверка роли для доступа к странице

if (

    $hash == $row['pwd'] &&

    ($row['role'] == 'admin' ||

    $row['role'] == 'moderator')

) {

    echo '<h2>Добро пожаловать! ;)</h2>';

} else {

    echo '<h2>Контент только для пользователей с правами
администратора или модератора...</h2>';

    echo '<p><a href="/">Выполнить вход</a></p>';

}

}

else {

    echo '<h2>Для просмотра контента необходима

```

```
аутентификация!</h2>';  
  
    echo '<p><a href="/">Выполнить вход</a></p>';  
  
}  
  
?>
```

example_4. admin/index.php

```
<?php  
  
    // если аутентификация состоялась  
  
    if (isset($_COOKIE['user'])) {  
  
        // подключение к базе данных  
  
        $mysqli = new mysqli ('localhost', 'root', '',  
            'db_auth');  
  
        // создание, выполнение подготовленного запроса  
  
        $stmt = $mysqli->prepare('SELECT * FROM `user_crypt`  
WHERE `login` = ?');  
  
        $stmt->bind_param('s', $_COOKIE['user']['login']);  
  
        $stmt->execute();  
  
        // получение результата  
  
        $res = $stmt->get_result();  
  
        $row = $res->fetch_assoc();  
  
        // получаем хеш пароля  
  
        $hash = crypt($_COOKIE['user']['pwd'], $row['salt']);  
  
        // проверка совпадения паролей  
  
        // проверка роли для доступа к странице  
  
        if (  
  
            $hash == $row['pwd'] &&  
  
            $row['role'] == 'admin'
```

```

    ) {

        echo '<h2>Добро пожаловать! ;) </h2>';

    } else {

        echo '<h2>Контент только для администраторов!</h2>';

        echo '<p><a href="/">Выполнить вход</a></p>';

    }

} else {

    echo '<h2>Для просмотра контента необходима
аутентификация!</h2>';

    echo '<p><a href="/">Выполнить вход</a></p>';

}

?>

```

Примечание. Не стоит информировать пользователя отдельными предупреждениями о неверно введенном пароле или логине. Достаточно предупредить, что пользователь с такими данными не найден.

Хеш, созданный функцией **crypt()** может быть также проверен функцией **password_verify()**. В демонстрационном примере **example_5** разберем применение **password_verify()** совместно с **crypt()**.

example_5. index.html

```

<form action="auth.php" method="post">

    Логин: <input type="text" name="login" value=""><p>

    Пароль: <input type="text" name="pwd" value=""><p>

    <button>Войти</button>

</form>

```

```
<p><a href='auth.php'>Пропустить аутентификацию</a></p>
```

example_5. auth.php

```
<?php

// если есть POST-данные формы

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

    // подключение к базе данных

    $mysqli = new mysqli ('localhost', 'root', '',
        'db_auth');

    // создание, выполнение подготовленного запроса

    $stmt = $mysqli->prepare('SELECT * FROM `user_crypt`
        WHERE `login` = ?');

    $stmt->bind_param('s', $_POST["login"]);

    $stmt->execute();

    // получение результата

    $result = $stmt->get_result();

    // если запрос вернул запись

    if ($result->num_rows) {

        $row = $result->fetch_assoc();

        // если пароли совпадают

        if (password_verify ($_POST['pwd'], $row['pwd'])) {

            $msg = "<h2>Поздравляю! Вы успешно
                аутентифицированы!</h2>";

            $msg .= "<p>Вход выполнил пользователь:
                {$_POST["login"]}</p>";

            // установка cookie

            setcookie('user[login]', $_POST["login"], 0,
                '/');
```

```

        setcookie('user[pwd]', $_POST['pwd'], 0, '/');

    }

    else $err = true;

}

else $err = true;

}

elseif (isset($_COOKIE['user']))

    $msg = "<p>Вы продолжаете работать как:
    {$_COOKIE['user']['login']}</p>";

else

    $msg = "<p>Выполнить <a href='/'>вход</a> вы можете в
    любое удобное для вас время</p>";

?>

<!-- ... -->

<!-- меню сайта -->

<p>

    <a href='/user/'>Посетителям</a> |

    <a href='/moderator/'>Модераторам</a> |

    <a href='/admin/'>Администраторам</a>

</p>

<?php

    if (isset($err)) {

        echo "<h3>Пользователь с такими данными в системе не
        найден</h3>";

        echo "<p><a href='/'>Попробовать еще раз</a></p>";

    }

    else echo $msg;

?>

```

example_5. user/index.php

```
<?php

// если аутентификация состоялась

if (isset($_COOKIE['user'])) {

    // подключение к базе данных

    $mysqli = new mysqli ('localhost', 'root', '',
        'db_auth');

    // создание, выполнение подготовленного запроса

    $stmt = $mysqli->prepare('SELECT * FROM `user_crypt`
        WHERE `login` = ?');

    $stmt->bind_param('s', $_COOKIE['user']['login']);

    $stmt->execute();

    // получение результата

    $res = $stmt->get_result();

    $row = $res->fetch_assoc();

    // проверка совпадения паролей

    // совпадения паролей достаточно для определения,
    // зарегистрирован ли пользователь

    if (

        password_verify ($_COOKIE['user']['pwd'],
            $row['pwd'])

    ) {

        echo '<h2>Для зарегистрированных пользователей
            презент! ;)</h2>';

    }

}
```

?>

example_5. moderator/index.php

```
<?php

// если аутентификация состоялась

if (isset($_COOKIE['user'])) {

    // подключение к базе данных

    $mysqli = new mysqli ('localhost', 'root', '',
        'db_auth');

    // создание, выполнение подготовленного запроса

    $stmt = $mysqli->prepare('SELECT * FROM `user_crypt`
        WHERE `login` = ?');

    $stmt->bind_param('s', $_COOKIE['user']['login']);

    $stmt->execute();

    // получение результата

    $res = $stmt->get_result();

    $row = $res->fetch_assoc();

    // проверка совпадения паролей

    // проверка роли для доступа к странице

    if (

        password_verify ($_COOKIE['user']['pwd'],
            $row['pwd']) &&

        ($row['role'] == 'admin' ||

            $row['role'] == 'moderator')

    ) {

        echo '<h2>Добро пожаловать! ;)</h2>';

    } else {

        echo '<h2>Контент только для пользователей с правами
            администратора или модератора...</h2>';

        echo '<p><a href="/">Выполнить вход</a></p>';
    }
}
```

```

    }

}

else {

    echo '<h2>Для просмотра контента необходима
аутентификация!</h2>';

    echo '<p><a href="/">Выполнить вход</a></p>';

}

?>

```

example_5. admin/index.php

```

<?php

// если аутентификация состоялась

if (isset($_COOKIE['user'])) {

    // подключение к базе данных

    $mysqli = new mysqli ('localhost', 'root', '',
'db_auth');

    // создание, выполнение подготовленного запроса

    $stmt = $mysqli->prepare('SELECT * FROM `user_crypt`
WHERE `login` = ?');

    $stmt->bind_param('s', $_COOKIE['user']['login']);

    $stmt->execute();

    // получение результата

    $res = $stmt->get_result();

    $row = $res->fetch_assoc();

    // проверка совпадения паролей

    // проверка роли для доступа к странице

    if (

        password_verify ($_COOKIE['user']['pwd'],

```

```
        $row['pwd']) &&

        $row['role'] == 'admin'

    ) {

        echo '<h2>Добро пожаловать! ;)</h2>';

    } else {

        echo '<h2>Контент только для администраторов!</h2>';

        echo '<p><a href="/">Выполнить вход</a></p>';

    }

} else {

    echo '<h2>Для просмотра контента необходима
аутентификация!</h2>';

    echo '<p><a href="/">Выполнить вход</a></p>';

}

?>
```

Следующая функция используется в реальных приложениях, про нее **нужно помнить**, но в первых самостоятельных приложениях использование функции не обязательно.

password_needs_rehash

password_needs_rehash() — проверяет, что указанный хеш соответствует заданным опциям.

Описание

```
password_needs_rehash($hash, $algo, $options = []);
```

Проверяет, что указанный хеш соответствует заданным опциям и заданному алгоритму. Если нет, то можно сделать вывод о том, что хеш надо пересоздать.

Параметры

- **hash** – хеш, созданный функцией **password_hash()**.
- **algo** – константа, обозначающая используемый алгоритм хеширования пароля.
- **options** – ассоциативный массив с опциями. Документацию по поддерживаемым опциям смотрите в открытых источниках.

Возвращаемые значения

Возвращает **true**, если пароль должен быть перехеширован с использованием алгоритма **algo** и опций **options**, или **false**, если нет.

Принцип использования функции **password_needs_rehash()** достаточно прозрачен.

example_6. index.html

```
<form action="auth.php" method="post">

    Логин: <input type="text" name="login" value=""><p>

    Пароль: <input type="text" name="pwd" value=""><p>

    <button>Войти</button>

</form>

<p><a href='auth.php'>Пропустить аутентификацию</a></p>
```

example_6. auth.php

```
<?php

// если есть POST-данные формы

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

    // подключение к базе данных

    $mysqli = new mysqli ('localhost', 'root', '',
        'db_auth');

    // создание, выполнение подготовленного запроса

    $stmt = $mysqli->prepare('SELECT * FROM `user_hash` WHERE
```

```

`login` = ?');

$stmt->bind_param('s', $_POST["login"]);

$stmt->execute();

// получение результата

$result = $stmt->get_result();

// если запрос вернул запись

if ($result->num_rows) {

    $row = $result->fetch_assoc();

    // если пароли совпадают

    if (password_verify ($_POST['pwd'], $row['pwd'])) {

        // если алгоритм хеширования поменялся

        if (password_needs_rehash($row['pwd'],
        PASSWORD_DEFAULT)) {

            // обновляем хеш

            $hash = password_hash($_POST['pwd'],
            PASSWORD_DEFAULT);

            $stmt = $mysqli->prepare('UPDATE `user_hash`
            SET `pwd` = ? WHERE `login` = ?');

            $stmt->bind_param('ss', $hash,
            $_POST["login"]);

            $stmt->execute();

        }

        $msg = "<h2>Поздравляю! Вы успешно
        аутентифицированы!</h2>";

        $msg .= "<p>Вход выполнил пользователь:
        {$_POST["login"]}</p>";

        // установка cookie

        setcookie('user[login]', $_POST['login'], 0,
        '/');
    }
}

```

```

        setcookie('user[pwd]', $_POST['pwd'], 0, '/');

    }

    else $err = true;

}

else $err = true;

}

elseif (isset($_COOKIE['user']))

    $msg = "<p>Вы продолжаете работать как:
    {$_COOKIE['user']['login']}</p>";

else

    $msg = "<p>Выполнить <a href='/'>вход</a> вы можете в
    любое удобное для вас время</p>";

?>

<!-- ... -->

<!-- меню сайта -->

<p>

    <a href='/user/'>Посетителям</a> |

    <a href='/moderator/'>Модераторам</a> |

    <a href='/admin/'>Администраторам</a>

</p>

<?php

    if (isset ($err)) {

        echo "<h3>Пользователь с такими данными в системе не
        найден</h3>";

        echo "<p><a href='/'>Попробовать еще раз</a></p>";

    }

    else echo $msg;

?>

```

example_6. user/index.php

```
<?php

// если аутентификация состоялась

if (isset($_COOKIE['user'])) {

    // подключение к базе данных

    $mysqli = new mysqli ('localhost', 'root', '',
        'db_auth');

    // создание, выполнение подготовленного запроса

    $stmt = $mysqli->prepare('SELECT * FROM `user_hash` WHERE
        `login` = ?');

    $stmt->bind_param('s', $_COOKIE['user']['login']);

    $stmt->execute();

    // получение результата

    $res = $stmt->get_result();

    $row = $res->fetch_assoc();

    // проверка совпадения паролей

    // совпадения паролей достаточно для определения,
    // зарегистрирован ли пользователь

    if (

        password_verify ($_COOKIE['user']['pwd'],
            $row['pwd'])

    ) {

        echo '<h2>Для зарегистрированных пользователей
            презент! ;)</h2>';

    }

}
```

?>

example_6. moderator/index.php

```
<?php

// если аутентификация состоялась

if (isset($_COOKIE['user'])) {

    // подключение к базе данных

    $mysqli = new mysqli ('localhost', 'root', '',
        'db_auth');

    // создание, выполнение подготовленного запроса

    $stmt = $mysqli->prepare('SELECT * FROM `user_hash` WHERE
        `login` = ?');

    $stmt->bind_param('s', $_COOKIE['user']['login']);

    $stmt->execute();

    // получение результата

    $res = $stmt->get_result();

    $row = $res->fetch_assoc();

    // проверка совпадения паролей

    // проверка роли для доступа к странице

    if (

        password_verify ($_COOKIE['user']['pwd'],
            $row['pwd']) &&

        ($row['role'] == 'admin' ||

            $row['role'] == 'moderator')

    ) {

        echo '<h2>Добро пожаловать! ;)</h2>';

    } else {

        echo '<h2>Контент только для пользователей с правами
            администратора или модератора...</h2>';

        echo '<p><a href="/">Выполнить вход</a></p>';

    }

}
```

```

    }

}

else {

    echo '<h2>Для просмотра контента необходима
аутентификация!</h2>';

    echo '<p><a href="/">Выполнить вход</a></p>';

}

?>

```

example_6. admin/index.php

```

<?php

// если аутентификация состоялась

if (isset($_COOKIE['user'])) {

    // подключение к базе данных

    $mysqli = new mysqli ('localhost', 'root', '',
'db_auth');

    // создание, выполнение подготовленного запроса

    $stmt = $mysqli->prepare('SELECT * FROM `user_hash` WHERE
`login` = ?');

    $stmt->bind_param('s', $_COOKIE['user']['login']);

    $stmt->execute();

    // получение результата

    $res = $stmt->get_result();

    $row = $res->fetch_assoc();

    // проверка совпадения паролей

    // проверка роли для доступа к странице

    if (

        password_verify ($_COOKIE['user']['pwd'],

```

```

        $row['pwd']) &&

        $row['role'] == 'admin'

    ) {

        echo '<h2>Добро пожаловать! ;) </h2>';

    } else {

        echo '<h2>Контент только для администраторов!</h2>';

        echo '<p><a href="/">Выполнить вход</a></p>';

    }

} else {

    echo '<h2>Для просмотра контента необходима
аутентификация!</h2>';

    echo '<p><a href="/">Выполнить вход</a></p>';

}

?>

```

Отмена аутентификации

Нашему приложению до сих пор не хватало возможности выполнения пользователем выхода из аккаунта – **logout**. В демо-примере **example_7** реализуем такую возможность.

Обратите внимание на вывод пункта меню в зависимости от наличия файла **cookie** для авторизованного пользователя.

```

<?php

// определяем пункт меню

if (isset($_COOKIE['user']))

    echo '<a href="/logout.php">Выход</a>';

else

    echo '<a href="/">Вход</a>';

```

example_7. index.html

```
<form action="auth.php" method="post">

    Логин: <input type="text" name="login" value=""><p>

    Пароль: <input type="text" name="pwd" value=""><p>

    <button>Войти</button>

</form>

<p><a href='auth.php'>Пропустить аутентификацию</a></p>
```

example_7. auth.php

```
<?php

    // если есть POST-данные формы

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

        // подключение к базе данных

        $mysqli = new mysqli ('localhost', 'root', '',
            'db_auth');

        // создание, выполнение подготовленного запроса

        $stmt = $mysqli->prepare('SELECT * FROM `user_hash` WHERE
            `login` = ?');

        $stmt->bind_param('s', $_POST["login"]);

        // получение результата

        $stmt->execute();

        $result = $stmt->get_result();

        // если запрос вернул запись

        if ($result->num_rows) {

            $row = $result->fetch_assoc();

            // если пароли совпадают

            if (password_verify($_POST['pwd'], $row['pwd'])) {
```

```

$msg = "<h2>Поздравляю! Вы успешно
аутентифицированы!</h2>";

$msg .= "<p>Вход выполнил пользователь:
{$_POST['login']}</p>";

// установка cookie

setcookie('user[login]', $_POST['login'], 0,
'/');

setcookie('user[pwd]', $_POST['pwd'], 0, '/');

    }

    else $err = true;

}

else $err = true;

}

elseif (isset($_COOKIE['user']))

    $msg = "<p>Вы продолжаете работать как:
    {$_COOKIE['user']['login']}</p>";

else

    $msg = "<p>Выполнить <a href='/'>вход</a> вы можете в
любое удобное для вас время</p>";

?>

<!-- ... -->

<!-- меню сайта -->

<p>

    <a href='/user/'>Посетителям</a> |

    <a href='/moderator/'>Модераторам</a> |

    <a href='/admin/'>Администраторам</a> |

<?php

    // определяем пункт меню

```

```

        if (isset($_COOKIE['user']))

            echo '<a href="/logout.php">Выход</a>';

        else

            echo '<a href="/">Вход</a>';

    ?>
</p>
<?php
    if (isset($err)) {

        echo "<h3>Пользователь с такими данными в системе не
        найден</h3>";

        echo "<p><a href='/'>Попробовать еще раз</a></p>";

    }

    else echo $msg;

?>

```

example_7. logout.php

```

<?php

    // удаляем cookie

    setcookie('user[login]', "", time()-3600, '/');

    setcookie('user[pwd]', "", time()-3600, '/');

    header('Location: /');

    exit;

```

example_7. user/index.php

```

<?php

    // если аутентификация состоялась

    if (isset($_COOKIE['user'])) {

```

```

        // подключение к базе данных

        $mysqli = new mysqli ('localhost', 'root', '',
        'db_auth');

        // создание, выполнение подготовленного запроса

        $stmt = $mysqli->prepare('SELECT * FROM `user_hash` WHERE
        `login` = ?');

        $stmt->bind_param('s', $_COOKIE['user']['login']);

        $stmt->execute();

        // получение результата

        $res = $stmt->get_result();

        $row = $res->fetch_assoc();

        // проверка совпадения паролей

        // совпадения паролей достаточно для определения,
        зарегистрирован ли пользователь

        if (

            password_verify($_COOKIE['user']['pwd'], $row['pwd'])

        ) {

            echo '<h2>Для зарегистрированных пользователей
            презент! ;)</h2>';

        }

    }

?>

```

example_7. moderator/index.php

```

<?php

    // если аутентификация состоялась

    if (isset($_COOKIE['user'])) {

        // подключение к базе данных
    }

```

```

$mysqli = new mysqli ('localhost', 'root', '',
'db_auth');

// создание, выполнение подготовленного запроса

$stmt = $mysqli->prepare('SELECT * FROM `user_hash` WHERE
`login` = ?');

$stmt->bind_param('s', $_COOKIE['user']['login']);

$stmt->execute();

// получение результата

$res = $stmt->get_result();

$row = $res->fetch_assoc();

// проверка совпадения паролей

// проверка роли для доступа к странице

if (

    password_verify($_COOKIE['user']['pwd'], $row['pwd'])
    &&

    ($row['role'] == 'admin' ||

    $row['role'] == 'moderator')

) {

    echo '<h2>Добро пожаловать! ;)</h2>';

} else {

    echo '<h2>Контент только для пользователей с правами
администратора или модератора...</h2>';

    echo '<p><a href="/">Выполнить вход</a></p>';

}

}

else {

    echo '<h2>Для просмотра контента необходима
аутентификация!</h2>';

```

```
        echo '<p><a href="/">Выполнить вход</a></p>';

    }

?>
```

example_7. admin/index.php

```
<?php

    // если аутентификация состоялась

    if (isset($_COOKIE['user'])) {

        // подключение к базе данных

        $mysqli = new mysqli ('localhost', 'root', '',
            'db_auth');

        // создание, выполнение подготовленного запроса

        $stmt = $mysqli->prepare('SELECT * FROM `user_hash` WHERE
            `login` = ?');

        $stmt->bind_param('s', $_COOKIE['user']['login']);

        $stmt->execute();

        // получение результата

        $res = $stmt->get_result();

        $row = $res->fetch_assoc();

        // проверка совпадения паролей

        // проверка роли для доступа к странице

        if (

            password_verify($_COOKIE['user']['pwd'], $row['pwd'])
            &&

            $row['role'] == 'admin'

        ) {

            echo '<h2>Добро пожаловать! ;) </h2>';

        }

    }

}
```

```

    } else {

        echo '<h2>Контент только для администраторов!</h2>';

        echo '<p><a href="/">Выполнить вход</a></p>';

    }

} else {

    echo '<h2>Для просмотра контента необходима
аутентификация!</h2>';

    echo '<p><a href="/">Выполнить вход</a></p>';

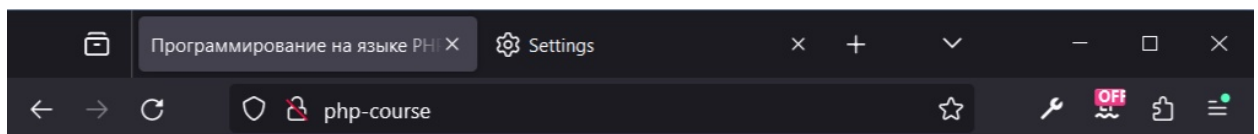
}

?>

```

В демонстрационном примере **example_8** в **cookie** сохраним не логин и пароль пользователя, а его **идентификатор (id)**.

Выполним некоторые изменения в логике работы приложения, добавим пункт меню, выведем дополнительную информацию о пользователе на главной странице приложения.



Аутентификация и авторизация с COOKIE

[Главная](#) | [Посетителям](#) | [Модераторам](#) | [Администраторам](#) | [Выход](#)

Главная страница сайта

Пользователь: **loginUser**

Вход выполнен с правами: **user**

example_8. index.php

```
<!-- меню сайта -->

<p>

    <a href="/">Главная</a> |

    <a href="/user/">Посетителям</a> |

    <a href="/moderator/">Модераторам</a> |

    <a href="/admin/">Администраторам</a> |

    <?php

        // определяем пункт меню

        if (isset($_COOKIE['user']))

            echo '<a href="/logout.php">Выход</a>';

        else

            echo '<a href="/form.html">Вход</a>';

    ?>

</p>

<h2>Главная страница сайта</h2>

<?php

    if (isset($_COOKIE['user'])):

        // подключение к базе данных

        $mysqli = new mysqli ('localhost', 'root', '',

            'db_auth');

        // создание, выполнение подготовленного запроса

        $stmt = $mysqli->prepare('SELECT * FROM `user_hash` WHERE

            `id_user` = ?');

        $stmt->bind_param('d', $_COOKIE['user']['id']);

        $stmt->execute();

        // получение результата
```

```

        $res = $stmt->get_result();

        $row = $res->fetch_assoc();

        echo "<p>Пользователь:<b> {$row['login']} </b></p>";

        echo "<p>Вход выполнен с правами:<b> {$row['role']}
        </b></p>";

    else:

        echo "<p>Для работы в системе необходима <a
        href='/form.html'>аутентификация</a></p>";

    endif;

?>

```

example_8. form.html

```

<form action="auth.php" method="post">

    Логин: <input type="text" name="login" value=""><p>

    Пароль: <input type="text" name="pwd" value=""><p>

    <button>Войти</button>

</form>

<p><a href='index.php'>Пропустить аутентификацию</a></p>

```

example_8. auth.php

```

<?php

    // подключение к базе данных

    $mysqli = new mysqli ('localhost', 'root', '', 'db_auth');

    // создание, выполнение подготовленного запроса

    $stmt = $mysqli->prepare('SELECT * FROM `user_hash` WHERE
    `login` = ?');

    $stmt->bind_param('s', $_POST["login"]);

```

```

// получение результата

$stmt->execute();

$result = $stmt->get_result();

// если запрос вернул запись

if ($result->num_rows) {

    $row = $result->fetch_assoc();

    // если пароли совпадают

    if (password_verify($_POST['pwd'], $row['pwd'])) {

        // установка cookie

        setcookie('user[id]', $row["id_user"], 0, '/');

        header('Location: /');

        exit;

    }

    else $err = true;

}

else $err = true;

?>

<!-- меню сайта -->

<p>

<a href= '/'>Главная</a> |

<a href= '/user/'>Посетителям</a> |

<a href= '/moderator/'>Модераторам</a> |

<a href= '/admin/'>Администраторам</a> |

<?php

    // определяем пункт меню

    if (isset($_COOKIE['user']))

        echo '<a href="/logout.php">Выход</a>';

```

```

        else

            echo '<a href="/form.html">Вход</a>';

        ?>
</p>
<?php

    if($err ?? false) {

        echo "<h3>Пользователь с такими данными в системе не
найден</h3>";

        echo "<p><a href='/form.html'>Попробовать еще
раз</a></p>";

    }

?>

```

example_8. logout.php

```

<?php

    // удаляем cookie

    setcookie('user[id]', '', time()-3600, '/');

    header('Location: /');

    exit;

```

example_8. user/index.php

```

<?php

    // если аутентификация состоялась

    if (isset($_COOKIE['user'])) {

        // подключение к базе данных

        $mysqli = new mysqli ('localhost', 'root', '',
'db_auth');

```

```

        // создание, выполнение подготовленного запроса

$stmt = $mysqli->prepare('SELECT * FROM `user_hash` WHERE
`id_user` = ?');

$stmt->bind_param('s', $_COOKIE['user']['id']);

$stmt->execute();

// получение результата

$res = $stmt->get_result();

// проверка наличия аутентифицированного пользователя

if ($res->num_rows) {

    echo '<h2>Для зарегистрированных пользователей
презент! ;)</h2>';

}

}

?>

```

example_8. moderator/index.php

```

<?php

// если аутентификация состоялась

if (isset($_COOKIE['user'])) {

    // подключение к базе данных

    $mysqli = new mysqli ('localhost', 'root', '',
'db_auth');

    // создание, выполнение подготовленного запроса

    $stmt = $mysqli->prepare('SELECT * FROM `user_hash` WHERE
`id_user` = ?');

    $stmt->bind_param('s', $_COOKIE['user']['id']);

    $stmt->execute();

    // получение результата

```

```

$res = $stmt->get_result();

$row = $res->fetch_assoc();

// проверка наличия аутентифицированного пользователя
// проверка роли для доступа к странице

if (

    $res->num_rows  &&

    ($row['role'] == 'admin' ||

    $row['role'] == 'moderator')

) {

    echo '<h2>Добро пожаловать! ;)</h2>';

} else {

    echo '<h2>Контент только для пользователей с правами
администратора или модератора...</h2>';

    echo '<p><a href="/form.html">Выполнить
вход</a></p>';

}

}

else {

    echo '<h2>Для просмотра контента необходима
аутентификация!</h2>';

    echo '<p><a href="/form.html">Выполнить вход</a></p>';

}

?>

```

example_8. admin/index.php

```
<?php
```

```
// если аутентификация состоялась

if (isset($_COOKIE['user'])) {

    // подключение к базе данных

    $mysqli = new mysqli ('localhost', 'root', '',
        'db_auth');

    // создание, выполнение подготовленного запроса

    $stmt = $mysqli->prepare('SELECT * FROM `user_hash` WHERE
        `id_user` = ?');

    $stmt->bind_param('s', $_COOKIE['user']['id']);

    $stmt->execute();

    // получение результата

    $res = $stmt->get_result();

    $row = $res->fetch_assoc();

    // проверка наличия аутентифицированного пользователя

    // проверка роли для доступа к странице

    if (

        $res->num_rows &&

        $row['role'] == 'admin'

    ) {

        echo '<h2>Добро пожаловать! ;)</h2>';

    } else {

        echo '<h2>Контент только для администраторов!</h2>';

        echo '<p><a href="/form.html">Выполнить
            вход</a></p>';

    }

} else {

    echo '<h2>Для просмотра контента необходима
        аутентификация!</h2>';

}
```

```
echo '<p><a href="/form.html">Выполнить вход</a></p>';

}

?>
```

Информацию о хеше можно получить, используя функцию **password_get_info()**.

password_get_info

password_get_info() — возвращает информацию о заданном хеше.

Описание

```
password_get_info($hash);
```

Если передан корректный хеш, созданный поддерживаемым **password_hash()** алгоритмом, то эта функция вернёт **информацию** об этом хеше.

Параметры

- **hash** – хеш, созданный функцией **password_hash()**.

Возвращаемые значения

Возвращает **ассоциативный массив** с тремя элементами:

- **algo** – содержит одну из констант алгоритмов паролей;
- **algoName** – содержит имя алгоритма в человекочитаемом виде;
- **options** – опции, переданные при вызове **password_hash()**.

На этом тему аутентификации с **cookie** будем считать завершённой.