Техническое задание

1. Главная страница сайта с каким-либо содержимым. Вверху страницы выводится:
    - если пользователь авторизован: *Добро пожаловать, %username%.*
    - если пользователь неавторизован: *Авторизация* — слово является ссылкой, которая ведёт на форму авторизации.
      Авторизован пользователь или нет, определяется с помощью cookie.
2. Страница с формой авторизации. Два инпута для логина и пароля и кнопкой «Вход». Если введены правильные логин и пароль, устанавливаются cookie со значениями переданных данных, а затем пользователя автоматически редиректит (перенаправляет) на главную страницу.
3. Страница для разлогинивания — при переходе на неё cookie будут удаляться из браузера пользователя, а затем выполняется редирект на главную страницу.

Реализация

1. Создаем простую базу данных с помощью массива, которую помещаем в файл usersDB.php.

```php
<?php

return [
    ['login' => 'admin', 'password' => 'P@sswOrd' ],
    ['login' => 'moderator', 'password' => 'word' ],
    ['login' => 'user', 'password' => '123' ]
];
```

2. Создадим файл auth.php и еапишем в нём функцию checkAuth, которая будет проверять, являются ли переданные в неё логин и пароль правильными. Для этого подключим файл с базой данных.

```php
<?php

function checkAuth(string $login, string $password){
    $users = require __DIR__ .'/usersDB.php';

    foreach ($users as $user){
        if($user['login'] === $login &&  $user['password'] === $password){
            return true;
        }
    }

    return false;
}
```

2.1. В функции checkAuth задаём аргументы $login и $password, тип устанавливаем как string.

2.2. В теле функции создаем переменную $users , в которую с помощью метода require помещаем подключение к нашему файлу с базой данных ($users = require __DIR__ .'/usersDB.php';). В результате получится массив $users с нашей базой данных.

2.3. Далее с помощью цикла foreach перебираем элементы массива $users. В массиве $users в данный момент есть три элемента (которые являются ассоциативными массивами. С ключами 'login' и 'password'.

2.3.1. Добавляем условие: если в массиве $users у ключа 'login' значение совпадает с тем, что ввёл пользователь в переменную $login (это то, что является аргументом функции checkAuth) И в массиве $users у ключа ' password ' значение совпадает с тем, что ввёл пользователь в переменную $password, то с помощью return возвращаем true.

3. В этом же файле auth.php напишем функцию **getUserLogin** , которая будет возвращать логин текущего пользователя. Эта функция проверит текущие значения cookie с ключами login и password с помощью уже существующей функции checkAuth. Если пользователь найдётся, то она вернёт его login, а иначе — null.

```php
function getUserLogin(): ?string  {
    $loginFromCookie = $_COOKIE['login']?? '';
    $passwordFromCookie = $_COOKIE['password']?? '';

    if(checkAuth($loginFromCookie, $passwordFromCookie) ){
        return $loginFromCookie;
    }

    return null;
}
```

3.1 Определяем функцию getUserLogin(). С помощью конструкции ": ?string"

Мы говорим, что типа для возвращаемых значений функции getUserLogin могут быть как строка, так и null.

3.2 В функции getUserLogin() определяем переменные $loginFromCookie и $passwordFromCookie, в которые с помощью глобального ассоциативного массива $_COOKIE, мы поместим ранее сохраненные куки (loginFromCookie = $_COOKIE['login']). Если такие сохранённые куки отсутствуют, то с помощью констукции "?? '' " говорим, что тогда в переменные $loginFromCookie и $passwordFromCookie помещаем пустые строки.

3.3 Добавляем условие если, в которую помещаем функцию checkAuth и в аргументы передаем переменные $loginFromCookie и $passwordFromCookie. Получается, что мы сначала передаем в эти переменные куки, если они есть, далее эти переменные с куки идут в функцию checkAuth, где идёт сверка действительно ли такие логин и пароль сущесвтуют в базе данных. Если существуют, то возвращаем в условии логин (return $loginFromCookie;), а если не существуют, то в результате работы функции getUserLogin() возвращается null.

4. Создаем файл index.php, где помещаем форму с приветствием для пользователей.

```php
<?php
require __DIR__.'/auth.php';
$login = getUserLogin();
?>

<html>
    <head>
        <title>Главная страница</title>
    </head>
    <body>
        <?php if($login !== null): ?>
            Добро пожаловать, <?= $login ?>
        <?php else: ?>
            <a href="/login.php">Авторизуйтесь</a>
        <?php endif; ?>
    </body>
</html>
```

4.1 Первым пунктом открываем php (<?php) и подключаем нашу проверку авторизации через cookie с помощью файла auth.php.

4.2 Создаем переменную $login, в которую помещаем результат работы функции getUserLogin(). Если функция находит куки для логина, то функция возвращает логин $loginFromCookie, а если нет, то вернёт null.

4.3 Пишем HTML код, где подключаем код php (здесь условия php написаны с помощью альтернативного синтаксиса – то есть вместо открывающейся и закрывающейся фигурных скобок после условия if() мы ставим двоеточие, а конце условия ставим ключевое слово endif).

4.3.1 Пишем условие: если в переменной $login значение не равно null , а это значит функция getUserLogin() нашла нужные куки, то приветствуем пользователя Добро пожаловать, <?= $login ?>.

4.3.2 Если переменная $login содержит null (то есть куки не найдены), отправляем пользователя с помощью ссылки <a href="/login.php"> на страницу авторизации login.php

5. Создаем файл login.php и пишем форму авторизации.

```php
<?php
if(!empty($_POST)){

    require __DIR__.'/auth.php';

    $login = $_POST['login'] ?? '';
    $password = $_POST['password'] ?? '';

    if(checkAuth($login, $password)){
        setcookie('login',$login, 0, '/');
        setcookie('password',$password, 0, '/');
        header ('Location: /index.php');
    }else{
        $error = 'Ошибка авторизации';
    }
}
?>
<html>
    <head>
        <title>Страница входа</title>
    </head>
    <?php if (isset($error)): ?>
    <span style="color: red" ><?= $error ?> </span>
    <?php endif; ?>
    <body>
        <form action="/login.php" method="post" >
            <input type="text" name="login" ><br>
            <input type="password" name="password" ><br>
            <input type="submit" value="Войти" >
        </form>
    </body>
</html>
```

5.1 Сначала напишем HTML (<html>,<head>,<title>).

5.1.1 Открываем <?php и добавляем условие, если (isset($error) – в переменной $error было установлено значение, отличное от null, то с помощью тега span и атрибута стиля выводим значение переменной $error красным цветом.

5.1.2 Далее создаем простую форму, где с помощью атрибута action страницу перенаправляем на саму себя (action="/login.php"), данные в форме отправляем методом post. В форме создаем две ячейки для логина и пароля кнопку “войти”.\

5.2 Перед HTML кода с формой добавляем код php.

5.2.1 Сначала добавляем условие проверки, есть ли данные в массиве POST. Проверка с помощью функции empty.

5.2.2 Если данные в массиве POST есть, то с помощью метода require подключаем файл с авторизацией пользователя (require __DIR__.'/auth.php';).

5.2.3 создаем переменные $login и $password и помещаем в них данные из массива POST (то что ввёл пользователь). Добавляем конструкцию ?? – если таких данных в массиве POST нет, то в переменные помещаем пустые строки.

5.2.4 Добавляем условие, в котором подключаем функцию checkAuth из файла с авторизацией auth.php. В функцию checkAuth в качестве аргументов мы помещаем переменные $login и $password (в которых должны быть либо данные, введённые пользователем в форме, либо пустые строки)

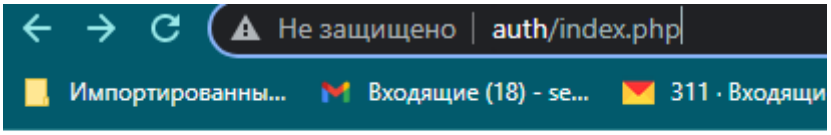
5.2.5 Далее с помощбю сетевой функции setcookie мы помещаем в параметр ‘login’ значение переменной $login (тоже самое делаем для пароля). Время жизни куки ставим 0 (то есть бесконечно время), и добавляем параметр ‘/’ что позволяет использовать куки на всём нашем сайте, а не только в файле login.php.

5.2.6 С помощью функции header() добавляем редирект на страницу index.php (header ('Location: /index.php');) – так после установки куки мы переходим на главную страницу.

5.2.7. Для условия if() добавляем else с выводом ошибки ($error = 'Ошибка авторизации';).

Тестирование

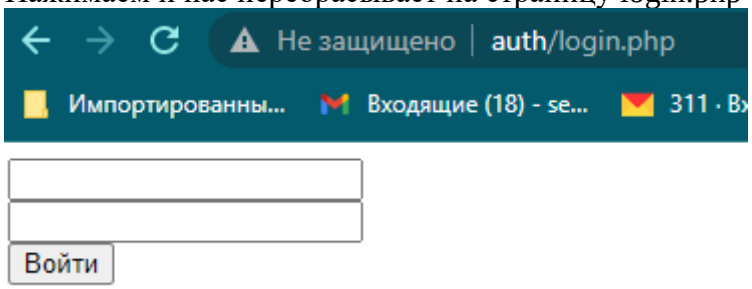
1. Заходим на страницу index.php



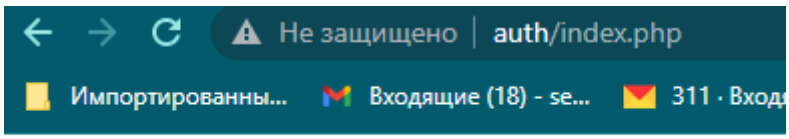
Авторизуйтесь

Где есть ссылка на авторизацию

2. Нажимаем и нас перебрасывает на страницу login.php

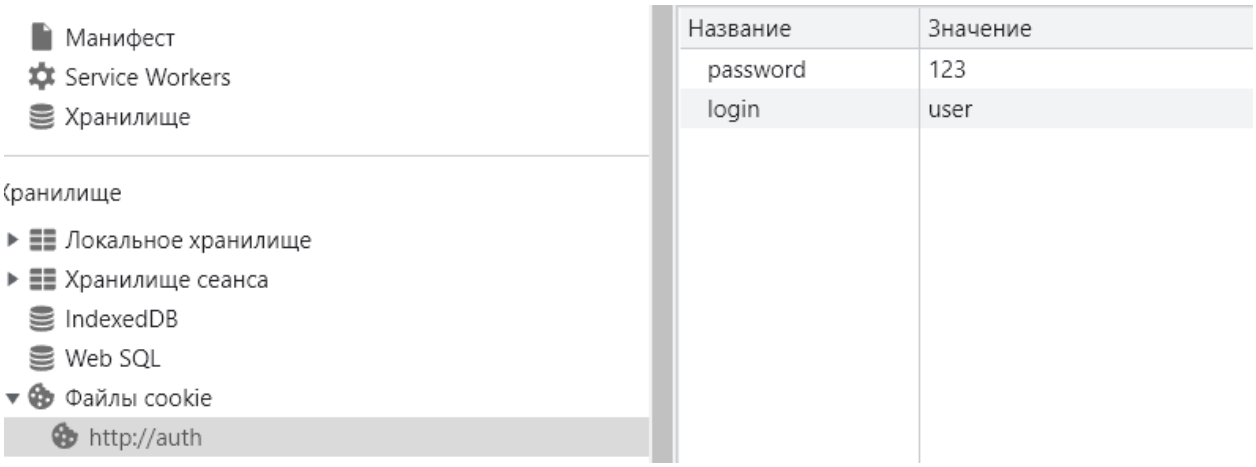


3. Вводим пару логин и пароль, которые существуют в нашей базе данных. User и пароль 123. И нас перенаправляет на главную страницу, где выводится приветствие.

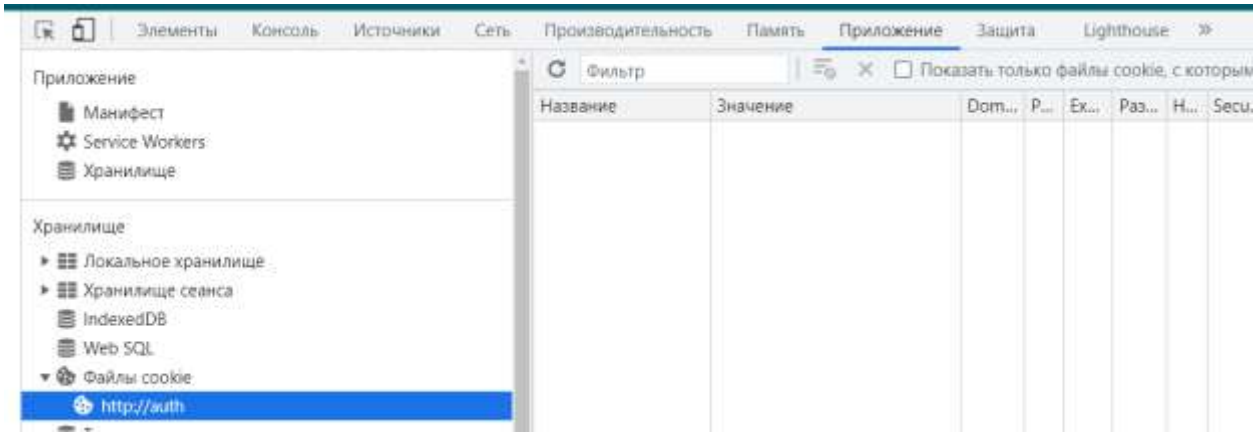


Добро пожаловать, user

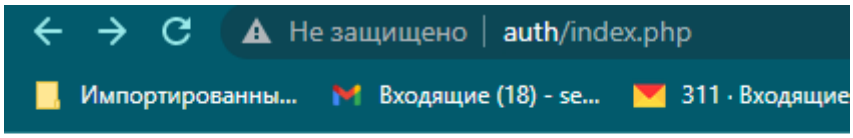
4. Смотрим что появилось в куки



5. Обновляем станицу index.php и видим, что также выведена надпись Добро пожаловать, user.
6. Остаёмся на этой странице и сбрасываем куки

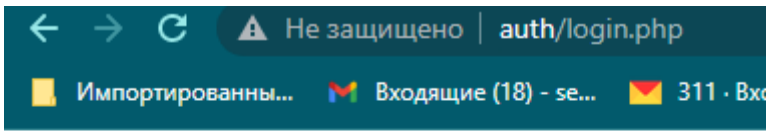


7. И опять обновляем страницу. Опять появилась ссылка для авторизации



Авторизуйтесь

8. Нажимаем на ссылку и попадаем на форму авторизации. Вводим к примеру логин user, но пароль заведомо неверный (например 45672). Вышла ошибка авторизации



Ошибка авторизации

Войти