

PHP. Уровень 2.

Разработка web-сайтов и взаимодействие с MySQL

Темы курса

- Использование cookie
- HTTP-заголовки ответа сервера
- Использование пользовательских сессий
- Работа с файловой системой
- Как отправить email?
- Основы работы с СУБД
- Сервер баз данных MySQL
- Использование MySQL в приложениях PHP
- Практическая работа: интернет-магазин

Модуль 1

РНР. Уровень 2

Использование cookie

Темы модуля

- Что такое cookie?
- Типы cookie
- Параметры cookie
- Создание cookie
- Чтение cookie
- Удаление cookie
- Дополнительные манипуляции с cookie

Как это работает?



```
POST /action.php HTTP/1.1
Host: www.example.com
...
name=Вася
```

```
HTTP/1.1 200 OK
Server: Microsoft IIS 7.5
...
Set-Cookie: UserName=Вася
<html> . . .
</html>
```

GET /folder/index.php HTTP/1.1 ↵

Host: www.example.com ↵

...

Cookie: UserName=Вася ↵



Манипуляции с cookie

```
// Создание временной cookie
setcookie("name", "John");

// Создание долговременной cookie на один час
setcookie("name", "John", time()+3600);

// Cookie доступны один час по пути /docs/
setcookie("name", "John", time()+3600, "/docs/");

// Cookie доступны один час по пути для всех поддоменов
setcookie("name", "John", time()+3600, "/", ".example.com");

// Cookie можно отдавать только при https
setcookie("name", "John", time()+3600, "/", ".example.com", true);

// Cookie можно отдавать только при http запросе
setcookie("name", "John", time()+3600, "/", ".example.com", false, true);

// Внимание: ошибка!
echo "Привет!";
setcookie("name", "John");

// Чтение cookie
echo $_COOKIE["name"];

// Удаление cookie
setcookie("name", "John", time()-3600);

// Массивы и cookie
$user = [
    'name' => 'John',
    'login' => 'root',
    'password' => '1234'
];
$str = serialize($user);
setcookie("user", $str);

$user = unserialize($_COOKIE["user"]);
print_r($user);

// Для сохранения целостности
$str = base64_encode(serialize($user));
setcookie("user", $str);
```

```
$user = unserialize( base64_decode($_COOKIE["user"]) );  
print_r($user);
```


Лабораторная работа 1

Использование cookie

Содержание лабораторной работы 1

Использование cookie

Упражнение 1: Создание и чтение cookie

- В текстовом редакторе откройте файл **inc\cookie.inc.php**
- Создайте целочисленную переменную **\$visitCounter** для подсчета количества посещений и присвойте ей значение по умолчанию **0**
- Проверьте, пришли ли куки по имени **visitCounter** от пользователя
- Если соответствующие куки пришли, то сохраните их значение в переменную **\$visitCounter**
- Увеличьте значение переменной **\$visitCounter** на единицу
- Создайте строковую переменную **\$lastVisit** для хранения значения последнего посещения страницы и присвойте ей значение по умолчанию **""** (пустая строка)
- Проверьте, пришли ли куки по имени **lastVisit** от пользователя
- Если соответствующие куки пришли, то сохраните их значение в переменную **\$lastVisit** отформатировав вывод даты
- Установите куки по имени **visitCounter** со значением равным значению переменной **\$visitCounter**
- Установите куки по имени **lastVisit** со значением равным текущей **временной метке** (timestamp)
- Сохраните файл **cookie.inc.php**

Упражнение 2: Вывод данных в браузер

- В текстовом редакторе откройте файл **index.php**
- В верхней части файла подключите файл **inc/cookie.inc.php**
- В блоке **<!-- Заголовок -->** внутри тэга **<blockquote>** выведите приветствие соблюдая условия:
 - Если посетитель зашел **первый раз**, то выведите:
Спасибо, что зашли на огонек
 - Если посетитель зашел **не первый раз**, то выведите:
Вы зашли к нам *такой-то* раз

Последнее посещение: *дата-время последнего посещения*

- Сохраните файл **index.php**
- Наберите в адресной строке браузера **localhost**
- Проверьте работу скрипта нажимая клавишу **[F5]**
- В файле **inc/cookie.inc.php** добавьте условие, чтобы куки устанавливались только один раз в день:
`if(date('d-m-Y', $_COOKIE['lastVisit']) != date('d-m-Y'))`

Что мы изучили?

- Уяснили механизм работы cookie
- Научились создавать временные и постоянные cookie
- Научились читать cookie через массив `$_COOKIE`

Модуль 2

РНР. Уровень 2

HTTP-заголовки ответа сервера.

Буферизация ответа.

Хэширование

Темы модуля

- HTTP-заголовки ответа сервера
- Переадресация
- Перезапрос ресурса
- Установка типа содержимого
- Управление кэшированием
- Буферизация вывода
- Хэширование

Переадресация и перезапрос ресурса

```
// Переадресация со статусом 302
header("Location: http://localhost");

// Переадресация со статусом 301
header("HTTP/1.1 301 Moved Permanently");
header("Location: http://localhost ");
// Или
header("Location: http://localhost ",true,301);

// Перезапрос ресурса
header("Refresh: 3");
header("Refresh: 3; url=http://localhost ");
```

Установка типа содержимого

```
// Принудительная установка типа передаваемого ресурса
header("Content-Type: text/xml");

// Принудительная установка кодировки передаваемого ресурса
header("Content-Type: text/html; charset=utf-8");

// Перенаправление вывода передаваемых данных
header("Content-Type: text/plain");
header("Content-Disposition: attachment; filename=\"myfile.txt
\"");
```


Управление кэшированием

```
// Запрет кэширования
header("Cache-Control: no-cache, max-age=0");

// Полный запрет кэширования
header("Cache-Control: no-store");

// Разрешение кэширования на один час относительно времени
запроса
header("Cache-Control: max-age=3600");

// Разрешение кэширования на один час
header("Expires: " . date("r", time() + 3600));
```

Буферизация вывода

```
// Включаем буферизацию
ob_start();
echo 'Hello world!';
setcookie("name", "John");
// Посылаем содержимое буфера
ob_flush();

echo 'Ещё контент!';
echo 'И ещё контент!';

// Посылаем содержимое буфера и отключаем его
ob_end_flush();

// Включаем буферизацию
ob_start();

echo "Hello ";
// Выбираем то, что находится в буфере. Его содержимое на
очищается!
$out1 = ob_get_contents();

echo "World";
// Выбираем то, что находится в буфере. Его содержимое на
очищается!
$out2 = ob_get_contents();

// А теперь очищаем буфер, но не закрываем
ob_clean();

echo "Саша";
echo " и ";
echo "Маша";

// Выбираем то, что находится в буфере
$out3 = ob_get_contents();

// Очищаем буфер и закрываем его
ob_end_clean();

echo $out1; // Hello
echo $out2; // Hello World
echo $out3; // Саша и Маша
```

Хэширование

```
echo md5("password");  
// 5f4dcc3b5aa765d61d8327deb882cf99  
  
echo sha1("password");  
// 5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8  
  
// Генерация хэша с солью  
$hash = crypt("password", '$1$password$');  
  
// Генерация стойкого хэша со случайной солью. Рекомендуется.  
$hash = password_hash("password", PASSWORD_BCRYPT);  
// Проверка пароля  
if( password_verify("password", $hash) )  
    echo "Пароль верный";
```

Что мы изучили?

- Научились гибко управлять браузером, посылая нужные заголовки ответа
- Включение буферизации хорошо сказывается на исполнении кода
- Все критичные данные необходимо хэшировать
- При хэшировании необходимо использовать "соль"
- При хэшировании никогда не нужно изобретать "велосипедов"

Модуль 3

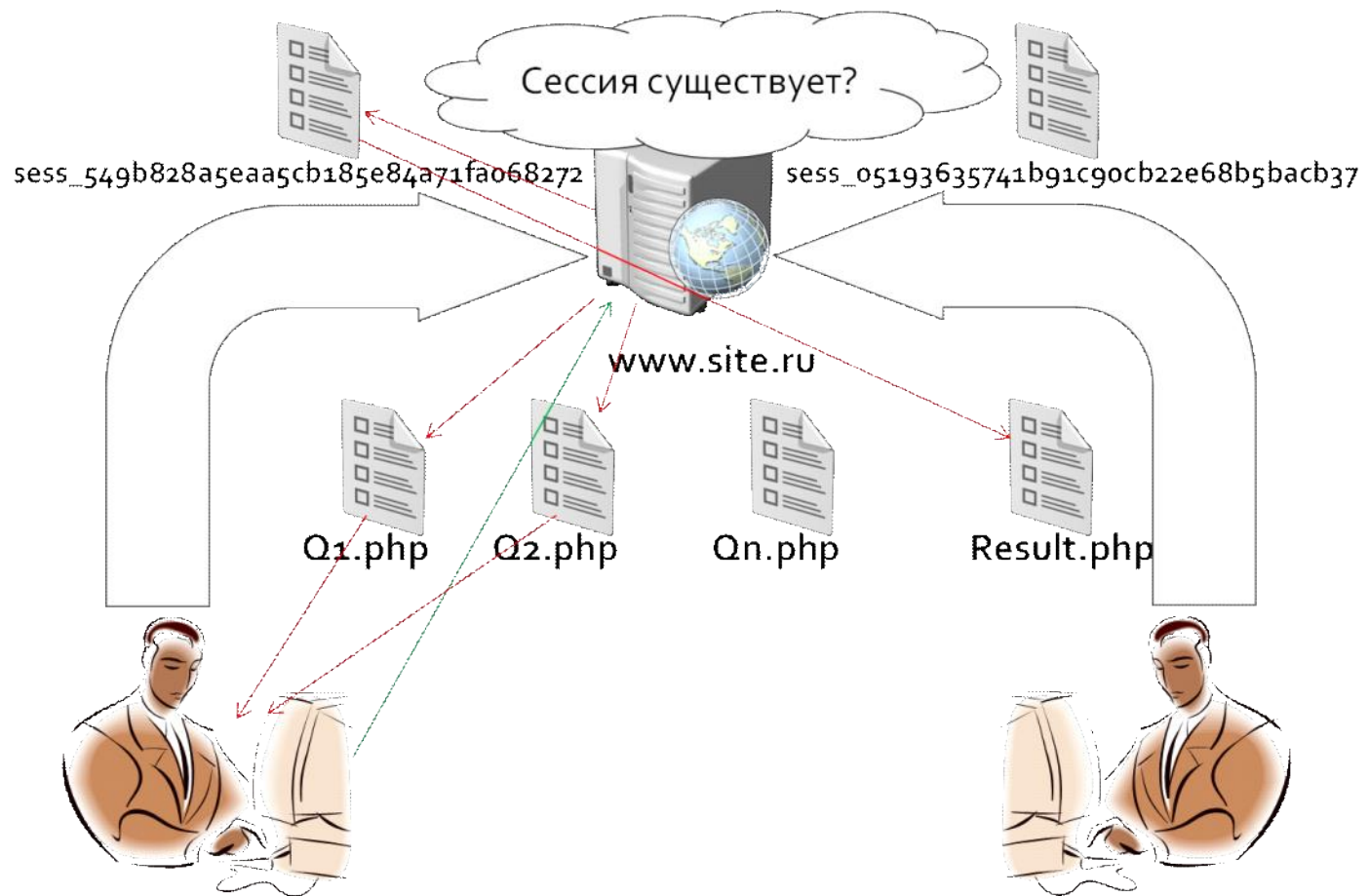
РНР. Уровень 2

Использование сеансов

Темы модуля

- Что такое сеанс (сессия)?
- Создание сессии
- Чтение сессии
- Удаление сессии
- Дополнительные параметры сессии

Как это работает?



Управление сеансами

```
// Создание и(или) доступ к сессии
session_start();

// Запись в сессионную переменную
$_SESSION['name'] = 'John';

// Чтение из сессионной переменной
echo $_SESSION['name'];

// Очистка сессионных переменных
session_destroy();

// Принудительное удаление сессионной cookie
setcookie(session_name(), session_id(), time()-3600);
```


Дополнительные параметры сеанса

```
// session.auto_start = 0
ini_set('session.name', 'PHPSESSID');
ini_set('session.save_path', '');
ini_set('session.gc_maxlifetime', '1440');
ini_set('session.cookie_lifetime', '0');
ini_set('session.cookie_httponly', '');
ini_set('session.cookie_path', '/');
```

Лабораторная работа 3

Использование сеансов

Содержание лабораторной работы 3

Использование сеансов

Упражнение 1: Основная настройка сеансов

- В текстовом редакторе откройте файл **test\index.php**
- В верхней части файла введите следующий текст:

```
<?php
session_start();
if(!isset($_SESSION['test']) and !isset($_POST['q'])) {
    // Если первый запуск теста, то инициализируем
    переменные
    $q = 0; // Номер текущего вопроса
    $title = 'Пройдите тест';
} else {
    // Создаем сессионную переменную test, содержащую массив
    ответов
    if($_POST['q'] != '1')
        $_SESSION['test'][] = $_POST['answer'];
    $q = $_POST['q'];
    $title = $_POST['title'];
}
?>
```
- В блоке **<!-- Верхняя часть страницы -->** между тэгами **<h1></h1>** введите следующий текст:

```
<?= $title?>
```
- В блоке **<!-- Область основного контента -->** введите следующий текст:

```
<?php
// В зависимости от номера вопроса,
// подключаем соответствующий файл с вопросами
switch($q){
    case 0:
        include 'start.php';
        break;
    case 1:
        include 'q1.php';
        break;
    case 2:
        include 'q2.php';
        break;
    case 3:
        include 'q3.php';
```

```

        break;
    default:
        include 'result.php';
    }
?>

```

- Сохраните файл **test\index.php**

Упражнение 2: Настройка страниц с вопросами

- Последовательно откройте в текстовом редакторе из папки **test** файлы **start.php**, **q1.php**, **q2.php** и **q3.php**, и пропишите в каждом из них внутри html-формы:
`<input type='hidden' name='q' value='<?= ++$q?>'>`
- Сохраните измененные файлы

Упражнение 3: Настройка вывода результата теста

- Откройте в текстовом редакторе файл **test\result.php**
- В верхней части файла введите следующий текст:

```

<?php
$result = 0; // Переменная для суммы ответов
if(isset($_SESSION['test'])){
    // Зачитываем ответы из ini-файла в массив
    $answers = parse_ini_file("answers.ini");
    // Проходим по ответам и смотрим, есть ли среди них
    правильные
    foreach($_SESSION['test'] as $value){
        if(array_key_exists($value, $answers))
            // Суммируем правильные ответы
            $result += (int)$answers[$value];
    }
    // Очищаем данные сессии
    session_destroy();
}
?>

```
- Добавьте в ячейку таблицы следующий текст:
`<p>Ваш результат: <?= $result?> из 30</p>`
- Сохраните файл **test\result.php**

Упражнение 4: Проверка работы он-лайн теста

- Запустите браузер и введите в адресной строке <http://localhost>
- Выберите в меню пункт Он-лайн тест

- Проверьте работоспособность он-лайн теста

Что мы изучили?

- Уяснили механизм работы сеансов
- Просто работаем с суперглобальным массивом `$_SESSION`
- Для 100% удаления сессии необходимо удалить сессионную cookie
- Есть возможность гибкой настройки сеансов с помощью дополнительных конфигурационных директив

Модуль 4

РНР. Уровень 2

**Операции с файлами и
директориями**

Темы модуля

- Функции общего назначения
- Функции для работы с потоком
- Функции для работы с файлами
- Функции для работы с директориями
- Загрузка файлов на сервер

Полезные функции

```
// Проверка файлов и директорий на существование
if( file_exists("file.txt") ){
    echo "Файл или директория существует";
}

if( is_file("file.txt") ){
    echo "Файл существует";
}

if( is_dir("images") ){
    echo "Директория существует";
}

// Размер файла
echo "Длина файла file.txt: " . filesize("file.txt");

// Время изменения и доступа к файлу
echo "Файл file.txt был изменён: " . filemtime("file.txt");
echo "и запрошен: " . fileatime("file.txt");

// Что можно с файлом делать?
if( is_readable("file.txt") ){
    echo "Файл можно читать";
}

if( is_writable("file.txt") ){
    echo "В файл можно писать";
}

if( is_executable("file.exe") ){
    echo "Файл можно исполнять";
}
```

Работаем с потоком

```
// Открытие потока на чтение и получение его дескриптора
$f = fopen("file.txt", "r") or die("Не могу открыть файл!");

// Закрытие потока
fclose($f);

// Открытие потока на чтение и запись
$f = fopen("file.txt", "r+");
// Открытие потока на запись. Указатель помещается в конец файла
$f = fopen("file.txt", "a");
// Открытие потока на чтение и запись. Указатель помещается в конец файла
$f = fopen("file.txt", "a+");
// Открытие потока на запись. Файл обрезается до нулевой длины
$f = fopen("file.txt", "w");
// Открытие потока на чтение и запись. Файл обрезается до нулевой длины
$f = fopen("file.txt", "w+");

// Читаем файл кусками
$f = fopen("file.txt", "r");
// Читаем первые 5 байт из потока
echo fread($f, 5);
// Читаем следующие 3 байта из потока
echo fread($f, 3);
// Выводим всё с текущей позиции до конца
fpassthru($f);
fclose($f);

// Читаем файл построчно в массив
$f = fopen("file.txt", "r");
$lines = [];
while ( $line = fgets($f) ){
    $lines[] = $line;
}
fclose($f);

// Читаем файл построчно в массив и вырезаем html-тэги, оставляя нужные
$f = fopen("file.html", "r");
$lines = [];
while ( $line = fgetss($f, 4096, "<p><br>") ){
    $lines[] = $line;
}
fclose($f);

// Читаем файл побайтово в массив
$f = fopen("file.txt", "r");
$bytes = [];
```

```
while ( !feof($f) ){
    $bytes[] = fgetc($f);
}
fclose($f);

// Пишем файл
$f = fopen("file.txt", "r+");
fwrite($f, "Новый текст", 25);
fclose($f);

// Пишем в конец файла
$f = fopen("file.txt", "a");
fputs($f, "\nНовая строка");
fclose($f);

// Читаем последние 10 байт из потока
$f = fopen("file.txt", "r");
// Устанавливаем указатель в нужную позицию
fseek($f, -10, SEEK_END);
// В какой позиции мы находимся?
echo ftell($f);
// Читаем данные
echo fread($f, 10);
// Устанавливаем указатель в начало потока
rewind($f);
fclose($f);
```

Прямая работа с файлами

```
// Читаем весь файл напрямую в буфер вывода
readfile("file.txt");
// Что и
$f = fopen("file.txt", "r");
echo fread($f, filesize("file.txt"));
fclose($f);

// Читаем файл построчно в массив
$lines = file("file.txt");;
// Что и
$f = fopen("file.txt", "r");
while ( $lines[] = fgets($f) );
fclose($f);

// Получаем весь файл в виде строки
$file = file_get_contents("file.txt");
// Что и
$f = fopen("file.txt", "r");
$file = fread($f, filesize("file.txt"));
fclose($f);

// Пишем в файл затирая содержимое
file_put_contents("file.txt", "Новое содержимое");
// Что и
$f = fopen("file.txt", "w");
fputs($f, "Новое содержимое");
fclose($f);

// Пишем в файл добавляя содержимое в конец
file_put_contents("file.txt", "Новое содержимое", FILE_APPEND);
// Что и
$f = fopen("file.txt", "a");
fputs($f, "Новое содержимое");
fclose($f);
```

Управление файлами

```
// Копируем файл
copy("source.txt", "destination.txt");

// Переименовываем файл
rename("old.txt", "new.txt");

// Удаляем файл
unlink("file-to-delete.txt");
```

Лабораторная работа 4

Создание и чтение журнала посещения страниц

Содержание лабораторной работы 4

Создание и чтение журнала посещения страниц

Упражнение 1: Уяснение задачи и основная настройка

- Необходимо отследить пути перемещение пользователя по сайту и записать их в файл из расчета: один путь - одна строка
- В текстовом редакторе откройте файл **index.php**
- В верхней части файла в php-блоке введите следующий текст:

```
// Имя файла журнала  
define('PATH_LOG', 'path.log');  
include 'inc/log.inc.php';
```
- Добавьте в навигационное меню пункт:

```
<li><a href='index.php?id=log'>Журнал посещений</a></li>
```
- Сохраните файл **index.php**

Упражнение 2: Создание файла журнала

- Откройте в текстовом редакторе файл **inc\log.inc.php**
- Создайте переменную **\$dt** и присвойте ей текущую **временную метку** (timestamp)
- Создайте переменную **\$page** и присвойте ей значение серверной переменной **REQUEST_URI**
- Создайте переменную **\$ref** и присвойте ей значение серверной переменной **HTTP_REFERER**
- Создайте переменную **\$path** и присвойте ей строку содержащую значения вышесозданных переменных. Строка должна иметь уникальный разделитель значений и выглядеть примерно так:
\$dt|\$page|\$ref
- Не забудьте указать в конце строки символ перевода на новую строку!
- Запишите сформированную строку в файл любым удобным для вас способом
- Файл журнала должен храниться в папке **log** под именем, определенным в константе **PATH_LOG**

- Сохраните файл **inc\log.inc.php**
- Запустите браузер и введите в адресной строке <http://localhost>
- Произвольно выбирая пункты меню **Домой, Контакты, О нас, Информация** и **Гостевая книга**, передвигайтесь по этим страницам
- Убедитесь, что данные записываются в файл в правильном формате

Упражнение 3: Вывод данных из файла журнала

- Откройте в текстовом редакторе файл **inc\view-log.inc.php**
- Проверьте, существует ли файл журнала?
- Если файл существует, получите все содержимое файла в виде массива строк
- В цикле выведите списком данные из файла журнала в произвольном виде, например:
21-06-2012 12:05:15 - /index.php?id=contacts ->
<http://localhost/>
- Сохраните файл **inc\view-log.inc.php**
- Запустите браузер и введите в адресной строке <http://localhost>
- Выберите пункт меню **Журнал посещений**
- Убедитесь, что данные корректно выводятся из файла журнала

Работа с директориями

```
// Создание директории
mkdir("newdir");
// Удаление директории
rmdir("dir-to-delete");
// Имя текущей директории
echo getcwd();

// Заходим в текущую директорию
$dir = opendir(".");
// Читаем содержимое директории
while ( $name = readdir($dir) ){
    if(is_dir($name))
        echo '[' . $name . ']<br>';
    else
        echo $name . '<br>';
}
// Выходим из директории
closedir($dir);

// Читаем содержимое директории в массив
$dir_content = scandir(".");

// Читаем определённое содержимое директории в массив
$dir_txt_content = glob("*.txt");
```

Загрузка файлов на сервер

```
<?php
/*
 * Настройки PHP.INI
 * file_uploads = "1"
 * upload_max_filesize = "2M"
 * post_max_size = "8M"
 * max_file_uploads = 20
 * upload_tmp_dir =
 * max_input_time = "-1"
 */
?>

<form enctype="multipart/form-data" method="POST" action="" >
    <input type="hidden" name="MAX_FILE_SIZE" value="4096" />
    <input name="userfile" type="file" />
    <input type="submit" />
</form>

<?php
if ( $_FILES["userfile"]["error"] != UPLOAD_ERR_OK ){

    switch($_FILES["userfile"]["error"]){
        case UPLOAD_ERR_INI_SIZE:
            echo "Превышен максимально допустимый размер"; break;
        case UPLOAD_ERR_FORM_SIZE:
            echo "Превышено значение MAX_FILE_SIZE"; break;
        case UPLOAD_ERR_PARTIAL:
            echo "Файл загружен частично"; break;
        case UPLOAD_ERR_NO_FILE:
            echo "Файл не был загружен"; break;
        case UPLOAD_ERR_NO_TMP_DIR:
            echo "Отсутствует временная папка"; break;
        case UPLOAD_ERR_CANT_WRITE:
            echo "Не удалось записать файл на диск";
        }
    }

}else{
    echo "Размер загруженного файла: " . $_FILES["userfile"]["size"];
    echo "Тип загруженного файла: " . $_FILES["userfile"]["type"];
    move_uploaded_file($_FILES["userfile"]["tmp_name"], "upload/" . $_FILES["userfile"]["name"]);
}
```

Что мы изучили?

- Научились работать с файлами и директориями
- Научились загружать файлы на сервер используя HTTP

Модуль 5

РНР. Уровень 2

Как отправить email?

Темы модуля

- Настройки для работы с почтой
- Принципы работы с почтой
- Дополнительные параметры

Отправляем email

```
// Директивы PHP.INI
ini_set("SMTP", "localhost");
ini_set("smtp_port", "25");
ini_set("sendmail_from", "");

$to = "vasya@mail.ru";
$subject = "Проба пера";
$body = "Отправляю письмо Васе";

if( mail($to, $subject, $body) )
    echo "Письмо отправлено";
else
    echo "Письмо отправить не удалось";

// Используем дополнительные заголовки
$headers = "Content-Type: text/html;charset=utf-8\r\n";
$headers .= "To: Петя <petya@mail.ru>\r\n";
$headers .= "Cc: lена@mail.ru\r\n";
$headers .= "Bcc: sveta@mail.ru\r\n";
$headers .= "From: Федя <fedya@mail.ru>\r\n";

$body = "<h1>Отправляю письмо Васе и Пете</h1>";

mail($to, $subject, $body, $headers);
```

Что мы изучили?

- Узнали, как отправить email встроенными в PHP средствами

Модуль 6

РНР. Уровень 2

Основы работы с СУБД

Темы модуля

- Общие сведения о СУБД
- Требования к структуре базы данных
- Общие сведения о SQL
- Основные манипуляции с данными

Реляционная СУБД

- Реляционная модель ориентирована на организацию данных в виде двумерных таблиц
- Каждая таблица имеет своё имя
- Таблицы состоят из столбцов и строк
- Каждый элемент таблицы — один элемент данных
- Все ячейки в столбце таблицы имеют одинаковый тип
- Каждый столбец имеет уникальное имя
- Одинаковые строки в таблице отсутствуют
- Порядок следования строк и столбцов может быть произвольным
- Запросы к базе данных возвращают результат в виде таблиц, которые тоже могут выступать как объект запросов

Требования к структуре базы данных

- Проектирование БД – создание эффективной структуры данных, обеспечивающее хранение требуемой информации
 - Хорошая структура
 - Максимально упрощает взаимодействие с БД
 - Гарантирует непротиворечивость данных
 - Выжимает максимум производительности из системы
 - Плохая структура
 - Приводит к непониманию результатов выполнения запросов
 - Повышает риск введения в БД противоречивой информации
 - Порождает избыточные данные
 - Усложняет выполнение изменений структуры созданных ранее и уже заполненных данных таблицы.
- Целостность базы данных (database integrity)
 - Соответствие имеющейся в базе данных информации её внутренней логике, структуре и всем явно заданным правилам.
- Нормализация
 - Процесс преобразования отношений базы данных к виду, отвечающему нормальным формам
- Нормальная форма

- Свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, которая потенциально может привести к логически ошибочным результатам выборки или изменения данных
- Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение
- Исключение некоторых типов избыточности
- Устранение некоторых аномалий обновления
- Разработка проекта базы данных, который интуитивно понятен и может служить хорошей основой для последующего расширения
- Упрощение процедуры применения необходимых ограничений целостности

Таблицы

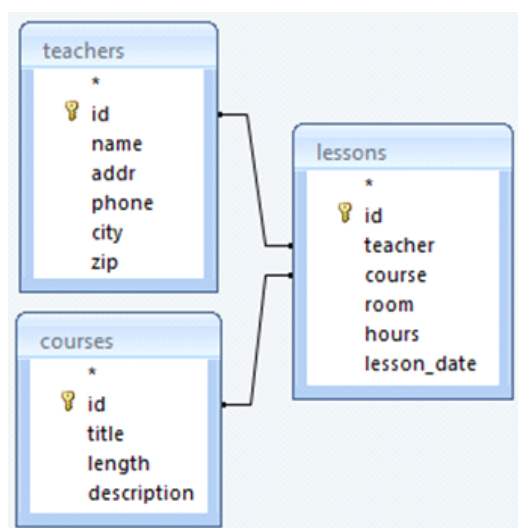
date	course	teacher	hourse
10/02/2008	HTML	Петров	8
10/02/2008	ASP	Иванов	4
11/03/2008	PHP	Иванов	8
12/04/2008	XML	Сидоров	4

- Таблица находится в первой нормальной форме, если каждый её атрибут атомарен и все строки различны. Под выражением «атрибут атомарен» понимается, что атрибут может содержать только одно значение.
- Таблица находится во второй нормальной форме, если она находится в первой нормальной форме, и при этом любой её атрибут, не входящий в состав первичного ключа, функционально полно зависит от первичного ключа

date	course	teacher	hourse
10/02/2008	1	1	8
10/02/2008	2	2	4
11/03/2008	3	2	8
12/04/2008	4	3	4

id	title
1	HTML
2	ASP
3	PHP
4	XML

id	name
1	Петров
2	Иванов
3	Сидоров



SQL

- SQL (англ. Structured Query Language — язык структурированных запросов)
 - универсальный язык, применяемый для создания, модификации и управления данными в реляционных базах данных
- Язык SQL делится на три части:
 - Операторы определения данных (Data Definition Language, DDL)
 - Операторы манипуляции данными (Data Manipulation Language, DML)
 - Операторы определения доступа к данным (Data Control Language, DCL)

Основные манипуляции с данными

Выборка данных

```
SELECT name FROM teachers
```

```
SELECT name, addr, city  
FROM teachers  
ORDER BY name
```

```
SELECT title  
FROM courses  
WHERE length > 30
```

```
SELECT *  
FROM courses  
WHERE length > 30  
AND title LIKE 'Web%'
```

```
SELECT DISTINCT length  
FROM courses
```

Объединение таблиц

id	name	code
1	Иванов	IVAN
2	Петров	PETR

id	tid	course
1	1	PHP
2	1	XML

```
SELECT t.name, t.code, l.course  
FROM teachers t  
INNER JOIN lessons l ON t.id = l.tid
```

name	code	course
Иванов	IVAN	PHP
Иванов	IVAN	XML

```
SELECT t.name, t.code, l.course
FROM teachers t
LEFT OUTER JOIN lessons l ON t.id = l.tid
```

name	code	course
Иванов	IVAN	PHP
Иванов	IVAN	XML
Петров	PETR	NULL

Вставка новой записи

```
INSERT INTO courses
VALUES (NULL, 'PHP', '...', 40)
```

```
INSERT INTO courses(title, length)
VALUES ('PHP', 40)
```

Удаление записи

```
DELETE FROM lessons
WHERE date = '2014-06-11'
```

Изменение записи

```
UPDATE teachers
SET
    zarplata = zarplata * 2,
    premia = premia * 10
WHERE name LIKE 'Иванов%'
OR name LIKE 'Петров%'
OR name LIKE 'Сидоров%'
```



```
UPDATE teachers
SET
    zarplata = zarplata * 2,
    premia = premia * 10
WHERE name IN
    ('Иванов', 'Петров', 'Сидоров')
```

Создание базы данных

```
CREATE DATABASE news
```

Создание таблицы и полей (СУБД MySQL диалект)

```
CREATE TABLE items (
    id int NOT NULL auto_increment,
    title varchar(255) NOT NULL default '',
    description varchar(255) NOT NULL default '',
    content text,
    author varchar(50) NOT NULL default '',
    pubdate timestamp NOT NULL default '',
    PRIMARY KEY (id)
)
```

Что мы изучили?

- Познакомились с понятием реляционных СУБД
- Уяснили требования, предъявляемые к реляционным СУБД
- Познакомились с основной терминологией
- Познакомились с основными SQL-операторами определения и манипуляции данными

Модуль 7

PHP. Уровень 2

Сервер баз данных MySQL

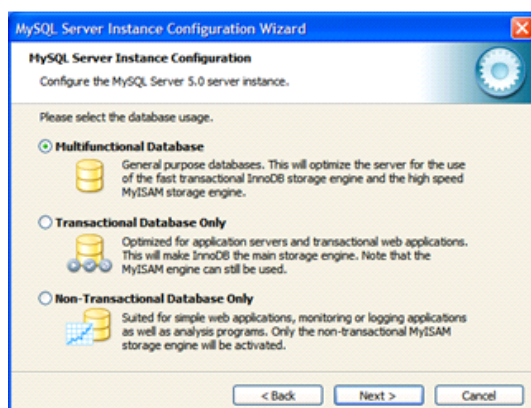
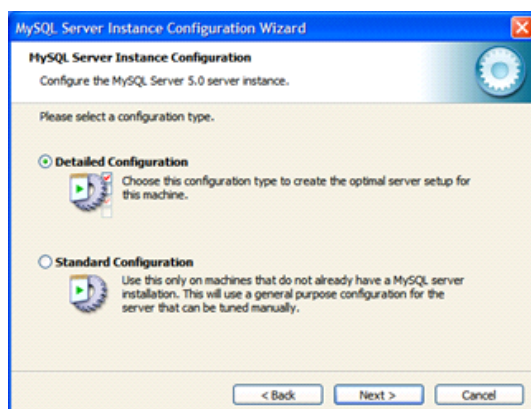
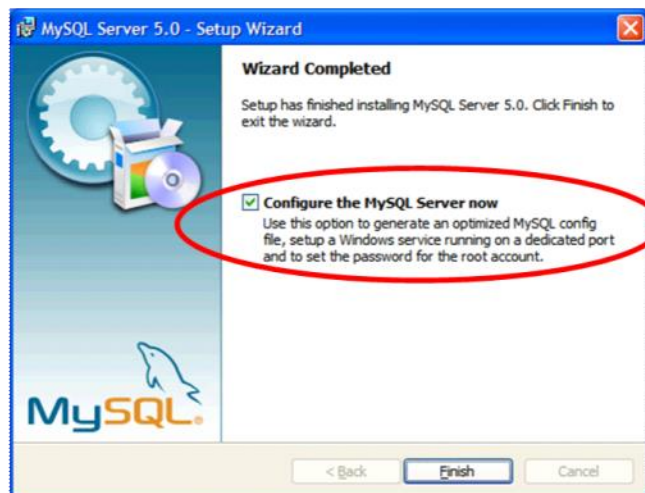
Темы модуля

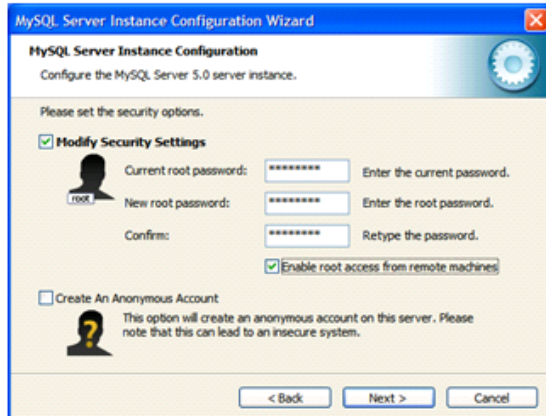
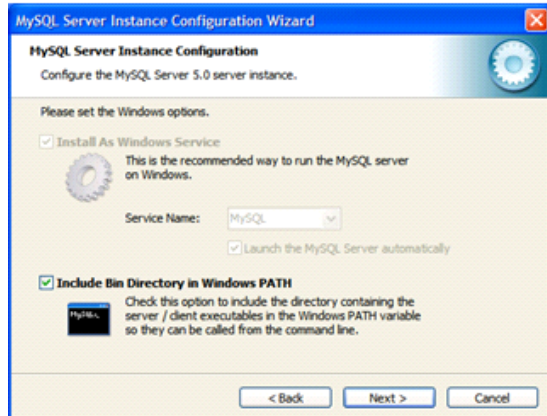
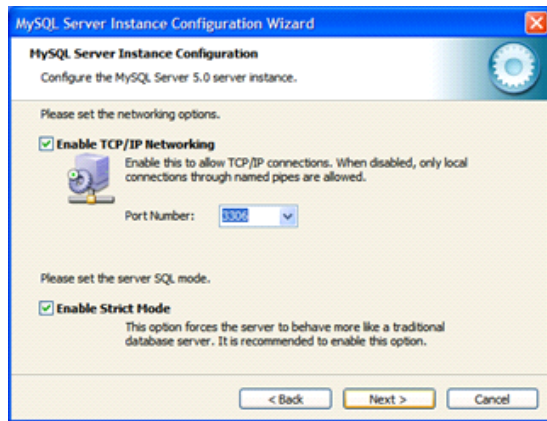
- Общие сведения о сервере баз данных MySQL
- Обзор установки сервера баз данных MySQL
- Обзор конфигурации сервера баз данных MySQL
- Программы для работы с сервером баз данных MySQL
- Использование утилит командной строки
- Экспорт и импорт баз данных

Общие сведения о СУБД MySQL

- Сервер баз данных MySQL – один из самых распространенных серверов баз данных
- <http://mysql.com>
- Основные параметры по-умолчанию
 - Порт: 3306
 - Логин: root
 - Пароль: -

Установка сервера баз данных MySQL





Конфигурация сервера баз данных MySQL?

- Конфигурационный файл **my.ini**

Программы для работы с СУБД MySQL

- GUI
 - MySQL Administrator
 - MySQL Query Browser
 - EMS MySQL Manager
 - другие
- Веб-приложение
 - www.phpmyadmin.net
- Утилиты командной строки
 - mysql.exe
 - mysqldump.exe

Использование утилит командной строки

- Соединение с монитором MySQL
 - `mysql -uлогин -рпароль`
- Внимание! Внутри монитора MySQL надо явно указывать конец запроса (;)
 - `USE имя_БД;`
 - `quit`
- Экспорт базы данных
 - `mysqldump -uлогин -рпароль имя_БД > dump.sql`
- Импорт базы данных
 - `mysql -uлогин -рпароль имя_БД < dump.sql`

Лабораторная работа 7

Создание базы данных и её использование

Содержание лабораторной работы 7

Создание базы данных и её использование

Инструкция по выполнению упражнений

- В конце каждой команды и запроса нажимайте клавишу **[ENTER]**
- В **мониторе MySQL** (только в нем) каждый запрос заканчивайте символом **точки с запятой (;)**
- В мониторе MySQL внимательно смотрите на результат, появляющийся после введённого запроса. В случае появления сообщения начинающегося со слова **ERROR**, вернитесь к предыдущему пункту и повторите его – где-то вы допустили ошибку. Не переходите к следующему пункту, пока не исправите ошибку!

Упражнение 1: Создание базы данных и её импорт из файла

- Наберите следующую команду:
mysql -uroot
- Вы попали в монитор MySQL. Последовательно наберите следующие запросы: **CREATE DATABASE web;**
quit
- Импортируйте данные из файла введя команду:
mysql -uroot web < web.sql
- В случае успешного выполнения последней команды никаких сообщений не будет! Только приглашение к вводу следующей команды

Упражнение 2: Использование полезных запросов

- В окне командной строки наберите команду: **mysql -uroot**
- Вы опять попали в монитор MySQL. Последовательно наберите

следующие запросы:

USE web;

SHOW TABLES;

и посмотрите, из каких таблиц состоит база данных **web**

- Введите запрос и проверьте число записей в таблице **courses**:
SELECT COUNT(*) FROM courses;
- Повторите предыдущий шаг для таблиц **lessons** и **teachers**
- Введите следующий запрос:
DESCRIBE courses;
и просмотрите структуру таблицы **courses**
- Повторите предыдущий шаг для таблиц **lessons** и **teachers**
- Введите следующий запрос:
SELECT * FROM teachers;
и просмотрите таблицы **teachers**
- Введите команду **quit** для выхода из монитора сервера MySQL

Что мы изучили?

- Получили информацию о том, как устанавливать сервер баз данных MySQL
- Рассмотрели основные настройки конфигурации сервера баз данных MySQL
- Научились использовать утилиту командной строки `mysql.exe`
- Научились основным манипуляциям с данными через монитор MySQL

Модуль 8

PHP. Уровень 2

Использование сервера баз данных MySQL в приложениях PHP

Темы модуля

- Расширение MySQLi
- Соединение с сервером баз данных MySQL
- Исполнение запросов
- Обработка результатов запросов
- Полезные функции
- Опасность SQL-инъекций
- Подготовленные запросы

Алгоритм работы с СУБД MySQL

- Подключение необходимого расширения в PHP.INI
- `php_mysql.dll`
- Установка соединения с сервером
- Выбор базы данных для работы (при необходимости)
- Исполнение запроса
- Обработка данных (при необходимости)
- Закрытие соединения

Соединение с сервером баз данных

```
// Соединение и выбор базы данных
$link = mysqli_connect('localhost', 'root', '', 'web');

// Отслеживаем ошибки при соединении
if( !$link ){
    echo 'Ошибка: '
        . mysqli_connect_errno()
        . ':'
        . mysqli_connect_error();
}

// Можно выбрать другую базу данных для работы
mysqli_select_db($link, 'test');

// Закрываем соединение
mysqli_close($link);
```

Основные манипуляции с сервером баз данных

```
// Соединение и выбор базы данных
$link = mysqli_connect('localhost', 'root', '', 'web');

// Посылаем простой запрос. Результат: true или false
$result = mysqli_query($link, "SET NAMES 'utf8'");

// Отслеживаем ошибки при исполнении запроса
if( !$result ){
    echo 'Ошибка: '
        . mysqli_errno($link)
        . ':'
        . mysqli_error($link);
}

// Посылаем запрос на выборку. Результат: object или false
$result = mysqli_query($link, 'SELECT * FROM teachers');

// Можно закрыть соединение
mysqli_close($link);

// Обрабатываем результат
$row = mysqli_fetch_array($result);
```

Массив		id	name	code
id	1	1	Иванов	IVAN
0	1	2	Петров	PETR
name	Иванов			
1	Иванов			
code	IVAN			
2	IVAN			

```
// Варианты обработки результата
// По-умолчанию
$row = mysqli_fetch_array($result);
$row = mysqli_fetch_array($result, MYSQLI_BOTH);

// Индексированный массив
$row = mysqli_fetch_row($result);
```

```
$row = mysqli_fetch_array($result, MYSQLI_NUM);

// Ассоциативный массив
$row = mysqli_fetch_assoc($result);
$row = mysqli_fetch_array($result, MYSQLI_ASSOC);

// Полная выборка: массив массивов
$row = mysqli_fetch_all($result, MYSQLI_ASSOC);
```

Полезные функции

```
$link = mysqli_connect('localhost', 'root', '', 'web');

// Экранируем строки!
$name = mysqli_real_escape_string($link, "John O'Brian");
$sql = "INSERT INTO teachers(name, email) VALUES('$name', 'johnh@gmail.com')";

mysqli_query($link, $sql);
// Получаем первичный ключ новой записи
$id = mysqli_insert_id($link);

$sql = "DELETE FROM lessons WHERE room = 'БК-1'";
mysqli_query($link, $sql);
// Сколько записей изменено?
$count = mysqli_affected_rows($link);

$sql = "SELECT * FROM courses";
$result = mysqli_query($link, $sql);
// Сколько записей вернулось?
$row_count = mysqli_num_rows($result);
// Сколько полей в вернувшихся записях?
$fields_count = mysqli_num_fields($result);
```

Лабораторная работа 8

Создание гостевой книги

Лабораторная работа 8

Создание гостевой книги

Упражнение 1: Создание базы данных и её структуры

- Наберите следующую команду:
`mysql -uroot`
- Последовательно наберите следующие запросы:
`CREATE DATABASE gbook;`
`quit`
- Импортируйте данные из файла введя команду:
`mysql -uroot gbook < gbook.sql`
- Войдите в монитор MySQL:
`mysql -uroot`
- Последовательно наберите следующие запросы:
`USE gbook;`
`DESCRIBE msgs;`
и изучите структуру таблицы `msgs`
- Не закрывайте окно командной строки! С его помощью вы будете отслеживать корректность исполняемых запросов

Упражнение 2: Основные настройки гостевой книги

- Откройте в текстовом редакторе файл `inc\gbook.inc.php`
- Выполняйте данное упражнение в блоке `/* Основные настройки */`
- Создайте константу `DB_HOST` со значением `localhost`
- Создайте константу `DB_LOGIN` со значением `root`
- Создайте константу `DB_PASSWORD` со значением `'пустая строка' ("")`
- Создайте константу `DB_NAME` со значением `gbook`
- Установите соединение с сервером базы данных MySQL, выбрав необходимую для работы базу данных

- Сохраните файл **inc\gbook.inc.php**

Упражнение 3: Добавление записей в базу данных

- Выполняйте данное упражнение в блоке **/* Сохранение записи в БД */**
- Проверьте, была ли отправлена веб-форма
- Примите и отфильтруйте полученные данные
- Сформируйте SQL-запрос на вставку данных в таблицу **msgs**. Запрос должен выглядеть примерно так:

```
INSERT INTO msgs (name, email, msg) VALUES ('$name', '$email', '$msg')
```
- Выполните запрос и отследите корректность его исполнения
- Сохраните файл **inc\gbook.inc.php**
- Запустите браузер и введите в адресной строке **http://localhost**
- Выберите пункт **Гостевая книга**
- Добавьте несколько записей, контролируя корректность исполнения запросов через монитор MySQL в окне командной строки

Упражнение 4: Вывод записей из базы данных

- Выполняйте данное упражнение в блоке **/* Вывод записей из БД */**
- Сформируйте SQL-запрос на выборку всех данных из таблицы **msgs** в обратном порядке. Запрос должен выглядеть примерно так:

```
SELECT id, name, email, msg, UNIX_TIMESTAMP(datetime) as dt
FROM msgs
ORDER BY id DESC
```
- Выполните запрос и результат выборки сохраните в переменной
- Закройте соединение с сервером базы данных MySQL
- Получите количество записей результата выборки и выведите примерно так:

```
<p>Всего записей в гостевой книге: количество записей</p>
```
- Используя цикл, выведите все сообщения, а также информацию об авторе каждого сообщения
- После каждого сообщения сформируйте ссылку для удаления этой записи. Информацию об идентификаторе удаляемого сообщения

передавайте методом **GET**

- В результате, каждая запись должна выглядеть примерно так:

```
<p>
  <a href="mailto:vasya@narod.ru">Вася Пупкин</a>
21-01-2015 в 13:45
  написал<br />Привет всем! Давайте дружить.
</p>
<p align="right">
  <a href="http://localhost    /index.php?id=gbook&del=1">
Удалить</a>
</p>
```
- Сохраните файл **inc\gbook.inc**
- Запустите браузер и введите в адресной строке <http://mysite.local>
- Выберите пункт **Гостевая книга**
- Убедитесь, что данные выводятся в правильном формате

Упражнение 5: Удаление записей из базы данных

- Выполняйте данное упражнение в блоке **/* Удаление записи из БД */**
- Проверьте, был ли отправлен запрос методом **GET** на удаление записи
- Примите и отфильтруйте полученные данные
- Сформируйте SQL-запрос на удаление записи. Запрос должен выглядеть примерно так:
DELETE FROM msgs WHERE id = \$del
- Выполните запрос и отследите корректность его исполнения
- Сохраните файл **inc\gbook.inc.php**
- Запустите браузер и введите в адресной строке **http://localhost**
- Выберите пункт **Гостевая книга**
- Убедитесь, что данные удаляются корректно

Опасность SQL-инъекций

```
/* Внедрение в строковые параметры */
// Запрос
$sql = "SELECT * FROM news WHERE title LIKE('%
$search_text%')";
// Инъекция
$search_text = "')+and+(author_id='1";
// Результат
$sql = "SELECT * FROM news WHERE title LIKE('%') AND
(author_id='1%')";

/* Расщепление запроса */
// Запрос
$sql = "SELECT * FROM news WHERE id = $id";
// Инъекция
$id = "12;INSERT INTO admin(login, password) VALUES('HaCkEr',
'1234')";
// Результат
$sql = "SELECT * FROM news WHERE id = 12;
        INSERT INTO admin(login, password) VALUES('HaCkEr',
'1234')";

/* Использование UNION */
// Запрос
$sql = "SELECT id, title, author, date FROM news WHERE id =
$id";
// Инъекция
$id = "-1 UNION SELECT 1, login, password, 1 FROM admin";
// Результат
$sql = "SELECT id, title, author, date FROM news WHERE id = -1
        UNION SELECT 1, login, password, 1 FROM admin";

/* Экранирование хвоста запроса */
// Запрос
$sql = "SELECT title, author
        FROM news WHERE cat = $id AND title LIKE('%
$search%')";
// Инъекция
$id = "-1 UNION SELECT login, password FROM admin--";
// Результат
$sql = "SELECT title, author FROM news WHERE cat = -1
        UNION SELECT login, password FROM admin--
        AND title LIKE('%$search%')";
```

```
/* Выполнение команд на сервере */  
// Запрос  
$sql = "SELECT name FROM user WHERE id = $id";  
// Инъекция  
$id = "1+UNION+SELECT+'<?eval($_GET[cmd]);?>'  
      +INTO+OUTFILE+' /www/cmd.php' --";  
// Исполнение скрипта  
// http://example.com?cmd=passthru\('ls'\)
```

Подготовленные запросы

```
$sql = "INSERT INTO users(name, email, age) VALUES(?, ?, ?)";
```

```
// Уважаемый сервер, вот запрос - разбери его
```

```
$stmt = mysqli_prepare($link, $sql);
```

```
// Уважаемый сервер, вот параметры для запроса
```

```
mysqli_stmt_bind_param($stmt, "ssi", $name, $email, $age);
```

```
// А теперь, исполни подготовленный запрос с переданными  
параметрами
```

```
mysqli_stmt_execute($stmt);
```

```
mysqli_stmt_close($stmt);
```

Что мы изучили?

- Научились использовать сервер баз данных MySQL в PHP-приложении
- Уяснили опасность SQL-инъекций
- Получили представление о способах борьбы с SQL-инъекциями

Модуль 9

РНР. Уровень 2

Практикум.

Создание интернет-магазина

Темы модуля

- Обзор архитектуры интернет-магазина
- Создание базы данных
- Создание каталога товаров
- Выборка и просмотр товаров из каталога
- Добавление товара в корзину
- Выборка просмотр товаров из корзины
- Удаление товаров из корзины
- Формирование заказа
- Выборка и просмотр заказов
- Повышение безопасности приложения (если останется время)

Архитектура интернет-магазина

- Каталог товаров (таблица **catalog**)
 - id (int, primary key)
 - title (string)
 - author (string)
 - pubyear (int)
 - price (int)

- Корзина покупателя (массив)
 - Хранится в cookie по имени "**basket**"
 - Инициализируется при первом обращении к приложению
 - Первый элемент массива всегда "**orderid**":
уникальный идентификатор заказа
 - Индекс последующих элементов - id товара
 - Значение последующих элементов -
количество товара
 - Пример: ["orderid"=>501bbdee1b4d3, 2
=>1, 6=>1]

- Список заказов (таблица **orders**)
 - id (int)
 - title (string)
 - author (string)
 - pubyear (int)
 - price (int)
 - quantity (int)
 - orderid (string)
 - datetime (int)

- Файл с персональными данными заказчика
orders.log

Структура приложения

- eshop.sql
 - дамп структуры базы данных
- inc\config.inc.php
 - основные настройки сайта
- inc\lib.inc.php
 - библиотека функций сайта
- admin\index.php
 - интерфейс админки
- admin\add2cat.php
 - HTML-форма для добавления товаров
- admin\save2cat.php
 - PHP-код обработки HTML-формы
- admin\orders.php
 - вывод списка заказов в виде HTML-таблицы
- admin\secure
 - папка с файлами для повышения безопасности приложения
- catalog.php
 - вывод списка товаров в виде HTML-таблицы
- add2basket.php
 - PHP-код обработки данных для добавления товара в корзину
- basket.php
 - вывод товаров покупателя в виде HTML-таблицы
- delete_from_basket.php
 - PHP-код обработки данных для удаления товара из корзины
- orderform.php
 - HTML-форма для добавления персональных данных покупателя
- saveorder.php

- РНР-код обработки персональных данных покупателя для сохранения в текстовый файл и перемещения товаров из корзины покупателя в таблицу

Практическая работа. Часть 1

Создание базы данных и её структуры

Содержание работы. Часть 1

Создание базы данных и её структуры

Задание 1: Создание базы данных и импорт таблиц из файла

- Наберите следующую команду:
mysql -uroot
- Последовательно наберите следующие запросы:
CREATE DATABASE eshop;
quit
- Импортируйте данные из файла введя команду:
mysql -uroot eshop < eshop.sql
- Войдите в монитор MySQL:
mysql -uroot
- Последовательно наберите следующие запросы:
USE eshop;
SHOW TABLES;
и изучите структуру базы данных **eshop**
- Используя SQL-оператор **DESCRIBE** последовательно просмотрите структуру таблиц базы данных **eshop**
- Не закрывайте окно командной строки! С его помощью вы будете отслеживать корректность исполняемых запросов

Практическая работа. Часть 2

Основные настройки приложения

Содержание работы. Часть 2

Основные настройки приложения

Задание 1: Объявление констант, переменных и установка соединения с сервером базы данных

- В текстовом редакторе откройте файл **eshop\inc\config.inc.php**
- Создайте константу **DB_HOST** со значением **localhost** для хранения адреса сервера баз данных MySQL
- Создайте константу **DB_LOGIN** со значением **root** для хранения логина соединения с сервером баз данных MySQL
- Создайте константу **DB_PASSWORD** со значением 'пустая строка' ("") для хранения пароля соединения с сервером баз данных MySQL
- Создайте константу **DB_NAME** со значением **eshop** для хранения имени базы данных
- Создайте константу **ORDERS_LOG** со значением **orders.log** для хранения имени файла с личными данными пользователей
- Создайте пустой массив **\$basket** для хранения корзины пользователя
- Создайте переменную **\$count** со значением 0 для хранения количества товаров в корзине пользователя
- Установите соединение с сервером базы данных MySQL, выбрав необходимую для работы базу данных. Не забудьте отследить возможную ошибку!
- Сохраните файл **eshop\inc\config.inc.php**

Практическая работа. Часть 3

Создание каталога товаров

Содержание работы. Часть 3

Создание каталога товаров

Задание 1: Создание функции добавления товара в каталог

- В текстовом редакторе откройте файл **eshop\inc\lib.inc.php**
- Создайте функцию **addItemToCatalog(\$title, \$author, \$pubyear, \$price)**, сохраняющую новый товар в таблицу **catalog** и принимающую в виде аргументов название, автора, год издания и цену товара
- В теле функции сформируйте подготовленный SQL-запрос на вставку данных в таблицу **catalog**. Запрос должен выглядеть примерно так:

```
$sql = 'INSERT INTO catalog (title, author, pubyear, price)
VALUES (?, ?, ?, ?)';
```
- Исполните подготовленный запрос:

```
if (!$stmt = mysqli_prepare($link, $sql))
    return false;
mysqli_stmt_bind_param($stmt, "ssii", $title, $author, $pubyear, $price);
mysqli_stmt_execute($stmt);
mysqli_stmt_close($stmt);
return true;
```
- Сохраните файл **eshop\inc\lib.inc.php**

Задание 2: Обработка полученных данных из веб-формы

- В текстовом редакторе откройте файл **eshop\admin\save2cat.php**
- Получите и отфильтруйте данные из формы
- Вызовите функцию **addItemToCatalog** для сохранения нового товара в базе данных:

```
if(!addItemToCatalog($title, $author, $pubyear, $price)){
    echo 'Произошла ошибка при добавлении товара в каталог';
}else{
    header("Location: add2cat.php");
    exit;
}
```
- Сохраните файл **eshop\admin\save2cat.php**

Задание 3: Добавление нового товара

- Откройте браузер и в адресной строке наберите:
http://localhost/eshop/admin
- Перейдите по ссылке **Добавление товара в каталог**. Вы должны попасть на страницу **add2cat.php**
- Добавьте в каталог несколько товаров
- За корректностью добавления следите через монитор MySQL в окне командной строки В случае ошибок, найдите их и исправьте

Практическая работа. Часть 4

Выборка и показ товаров из каталога

Содержание работы. Часть 4

Выборка и показ товаров из каталога

Задание 1: Создание функции выборки товара из каталога

- В текстовом редакторе откройте файл **eshop\inc\lib.inc.php**
- Создайте функцию **selectAllItems()**, которая возвращает все содержимое каталога товаров в виде ассоциативного массива
- В теле функции сформируйте SQL-запрос на выборку данных из таблицы **catalog**. Запрос должен выглядеть примерно так:

```
$sql = 'SELECT id, title, author, pubyear, price FROM catalog';
```
- Далее, напишите следующий текст:

```
if(!$result = mysqli_query($link, $sql))  
    return false;  
$items = mysqli_fetch_all($result, MYSQLI_ASSOC);  
mysqli_free_result($result);  
return $items;
```
- Сохраните файл **eshop\inc\lib.inc.php**

Задание 2: Формирование представления каталога

- В текстовом редакторе откройте файл **eshop\catalog.php**
- С помощью функции **selectAllItems** получите выборку всех товаров в переменную **\$goods**
- В цикле выведите все товары на экран:

```
foreach($goods as $item){  
    ?>  
    <tr>  
        <td><?= $item['title']?></td>  
        <td><?= $item['author']?></td>  
        <td><?= $item['pubyear']?></td>  
        <td><?= $item['price']?></td> <td>В корзину</td>  
    </tr>  
    <?>  
}
```
- Значение ячейки В корзину оформите в виде гиперссылки на документ **add2basket.php**, добавив параметр **id** с идентификатором товара

- Сохраните файл **eshop\catalog.php**
- Откройте браузер и в адресной строке наберите:
http://localhost/eshop/catalog.php
- Убедитесь в корректном выводе каталога товаров. В случае ошибок, найдите их и исправьте

Практическая работа. Часть 5

Сохранение товара в корзину пользователя

Содержание работы. Часть 5

Сохранение товара в корзину пользователя

Задание 1: Создание корзины покупателя

- В текстовом редакторе откройте файл **eshop\inc\lib.inc.php**
- Создайте функцию **saveBasket()**, которая сохраняет корзину с товарами в куки
- В теле функции введите следующий текст:

```
global $basket;  
$basket = base64_encode(serialize($basket));  
setcookie('basket', $basket, 0x7FFFFFFF);
```
- Создайте функцию **basketInit()**, которая создает либо загружает в переменную **\$basket** корзину с товарами, либо создает новую корзину с идентификатором заказа
- В теле функции введите следующий текст:

```
global $basket, $count;  
if(!isset($_COOKIE['basket'])){  
    $basket = ['orderid' => uniqid()];  
    saveBasket();  
}else{  
    $basket = unserialize(base64  
_decode($_COOKIE['basket']));  
    $count = count($basket) - 1;  
}
```
- Сохраните файл **eshop\inc\lib.inc.php**
- В текстовом редакторе откройте файл **eshop\inc\config.inc.php**
- Внизу кода допишите вызов функции **basketInit**
- Сохраните файл **eshop\inc\config.inc.php**

Задание 2: Создание функции сохранения товара в корзину

- В текстовом редакторе откройте файл **eshop\inc\lib.inc.php**
- Создайте функцию **add2Basket(\$id)**, которая добавляет товар в корзину пользователя и принимает к качеству аргумента идентификатор товара
- В теле функции введите следующий текст:

```
global $basket;  
$basket[$id] = 1;  
saveBasket();
```

- Сохраните файл **eshop\inc\lib.inc.php**

Задание 3: Добавление товара в корзину

- В текстовом редакторе откройте файл **eshop\add2basket.php**
- Получите идентификатор товара, добавляемого в корзину
- Назначьте количество добавляемого товара равным **1**
- Вызовите функцию **add2Basket** для сохранения товара в корзине
- Переадресуйте пользователя на страницу каталога товаров
- Сохраните файл **eshop\add2basket.php**
- Откройте браузер и в адресной строке наберите:
http://localhost/eshop/catalog.php
- Добавьте несколько товаров в корзину. В случае ошибок, найдите их и исправьте.
(Обратите внимание, что в строке **Товаров в корзине:** число должно меняться в большую сторону)

Практическая работа. Часть 6

Выборка и показ товаров из корзины пользователя

Содержание работы. Часть 6

Выборка и показ товаров из корзины пользователя

Задание 1: Создание функций выборки товаров из корзины

- В текстовом редакторе откройте файл **eshop\inc\lib.inc.php**
- Создайте функцию **myBasket()**, которая возвращает всю пользовательскую корзину в виде ассоциативного массива
- В теле функции введите следующий текст:

```
global $link, $basket;  
$goods = array_keys($basket);  
array_shift($goods);  
if(!$goods)  
    return false;  
$ids = implode(",", $goods);  
$sql = "SELECT id, author, title, pubyear, price  
        FROM catalog WHERE id IN ($ids)";  
if(!$result = mysqli_query($link, $sql))  
    return false;
```
- Создайте функцию **result2Array(\$data)**, которая принимает результат выполнения функции **myBasket** и возвращает ассоциативный массив товаров, дополненный их количеством
- В теле функции введите следующий текст:

```
global $basket;  
$arr = [];  
while($row = mysqli_fetch_assoc($data)){  
    $row['quantity'] = $basket[$row['id']];  
    $arr[] = $row;  
}  
return $arr;
```
- В теле функции **myBasket** допишите:

```
$items = result2Array($result);  
mysqli_free_result($result);  
return $items;
```
- Сохраните файл **eshop\inc\lib.inc.php**

Задание 2: Формирование корзины пользователя

- В текстовом редакторе откройте файл **eshop\basket.php**
- Проверьте, есть ли товары в корзине пользователя. Если товаров нет,

перед кодом отрисовки таблицы выведите сообщение: **Корзина пуста! Вернитесь в каталог.** Иначе, выведите сообщение: **Вернуться в каталог**

- Оформите слово каталог в виде гиперссылки на файл **catalog.php**
- Получите все товары из корзины пользователя в виде массива вызвав функцию **myBasket**
- Создайте целочисленные переменные **\$i** со значением **1** для подсчета порядковых номеров и **\$sum** со значением **0** для подсчета общей суммы заказа
- В цикле заполните таблицу по аналогии с заполнением каталога (Практическая работа 4, Задание 2)
- В цикле увеличивайте значение переменной **\$sum** на соответствующее значение (сумма текущего товара * количество товара) и значение переменной **\$i** на единицу
- Значение ячейки Удалить оформите в виде гиперссылки на документ **delete_from_basket.php**, добавив параметр **id** с идентификатором товара
- Выведите общую сумму товаров в корзине
- Сохраните файл **eshop\basket.php**
- Откройте браузер и в адресной строке наберите:
http://localhost/eshop/catalog.php
- Перейдите по ссылке в корзину и убедитесь в корректности вывода. В случае ошибок, найдите их и исправьте

Практическая работа. Часть 7

Удаление товара из корзины пользователя

Содержание работы. Часть 7

Удаление товара из корзины пользователя

Задание 1: Создание функции удаления товара из корзины

- В текстовом редакторе откройте файл **eshop\inc\lib.inc.php**
- Создайте функцию **deleteItemFromBasket(\$id)**, которая удаляет товар из корзины, принимая в качестве аргумента его идентификатор
- В теле функции удалите необходимый элемент массива **\$basket** и пересохраните корзину в куки
- Сохраните файл **eshop\inc\lib.inc.php**

Задание 2: Удаление товара из корзины пользователя

- В текстовом редакторе откройте файл **eshop\delete_from_basket.php**
- Получите идентификатор удаляемого товара
- Вызовите функцию **deleteItemFromBasket** для данного товара
- Переадресуйте пользователя на страницу корзины
- Откройте браузер и в адресной строке наберите:
http://localhost/eshop/catalog.php
- Перейдите по ссылке в корзину и попробуйте удалить товары. В случае ошибок, найдите их и исправьте
(Обратите внимание, что на странице каталога в строке **Товаров в корзине**: число должно меняться в меньшую сторону)

Практическая работа. Часть 8

Формирование заказа

Содержание работы. Часть 8

Формирование заказа

Задание 1: Создание файла с персональными данными пользователя

- В текстовом редакторе откройте файл **eshop\saveorder.php**
- Получите из веб-формы и обработайте данные заказа
- Получите идентификатор заказа
- Получите дату/время заказа в виде **временной метки** (timestamp)
- Создайте переменную **\$order**
- Присвойте переменной **\$order** строку состоящую из полученных данных, разделяя их символом "|" по следующему шаблону:
имя покупателя|email покупателя|телефонный номер покупателя|адрес покупателя|идентификатор заказа|дата/время заказа,
например:
"Иван Иванов|ivan@mail.ru|123-12-23|Москва, Сумской пр-д 17 кв.105|501bbdee1b4d3|1234567890"
(ВНИМАНИЕ: новые данные должны записываться в конец файла на новой строке!)
- Запишите значение переменной **\$order** в файл, имя которого хранится в константе **ORDERS_LOG**. Файл должен храниться в папке **eshop\admin!**
- Сохраните файл **eshop\saveorder.php**
- Откройте браузер и в адресной строке наберите:
http://localhost/eshop/catalog.php
- Перейдите по ссылке в корзину и нажмите кнопку **Оформить заказ!**
- Заполните данные веб-формы и нажмите кнопку **Заказать**
- Убедитесь, что нужный файл создается и данные записываются корректно. В случае ошибок, найдите их и исправьте
- Удалите созданный тестовый файл с данными пользователя

Задание 2: Создание функции добавления заказа в базу данных

- В текстовом редакторе откройте файл **eshop\inc\lib.inc.php**
- Создайте функцию **saveOrder(\$datetime)**, которая пересохраняет товары из корзины в таблицу базы данных **orders** и принимает в качестве аргумента **дату и время заказа** в виде временной метки
- В теле функции введите следующий текст:


```
global $link, $basket;
$goods = myBasket();
$stmt = mysqli_stmt_init($link);
$sql = 'INSERT INTO orders (
                                title,
                                author,
                                pubyear,
                                price,
                                quantity,
                                orderid,
                                datetime)
        VALUES (?, ?, ?, ?, ?, ?, ?)';
if (!mysqli_stmt_prepare($stmt, $sql))
    return false;
foreach($goods as $item){
    mysqli_stmt_bind_param($stmt, "ssiiisi",
                            $item['title'], $item['author'],
                            $item['pubyear'], $item['price'],
                            $item['quantity'],
                            $basket['orderid'],
                            $datetime);
    mysqli_stmt_execute($stmt);
}
mysqli_stmt_close($stmt);
return true;
```
- Перед строкой **return true;** удалите куки пользователя
- Сохраните файл **eshop\inc\lib.inc.php**

Задание 3: Создание заказа

- В текстовом редакторе откройте файл **eshop\saveorder.php**
- Вызовите функцию **saveOrder**, передав ей в качестве аргумента ранее полученную временную метку заказа
- Сохраните файл **eshop\saveorder.php**
- Откройте браузер и в адресной строке наберите:

http://localhost/eshop/catalog.php
- Перейдите по ссылке в корзину и нажмите кнопку **Оформить заказ!**

- Заполните данные веб-формы и нажмите кнопку **Заказать**
- Убедитесь в корректности записи данных в таблицу базы данных **orders**. В случае ошибок, найдите их и исправьте

Практическая работа. Часть 9

Просмотр списка заказов

Содержание работы. Часть 9

Просмотр списка заказов

Задание 1: Создание функции для выборки всех заказов

- В текстовом редакторе откройте файл **eshop\inc\lib.inc.php**
- Создайте функцию **getOrders()**, которая возвращает многомерный массив с информацией о всех заказах, включая персональные данные покупателя и список его товаров
- В теле функции введите следующий текст (комментарии можно опустить):

```
global $link;
if(!is_file(ORDERS_LOG))
    return false;
/* Получаем в виде массива персональные данные пользователей из файла */
$orders = file(ORDERS_LOG);
/* Массив, который будет возвращен функцией */
$allorders = [];
foreach ($orders as $order) {
    list($name, $email, $phone, $address, $orderid, $date) = explode("|",
$order);
    /* Промежуточный массив для хранения информации о конкретном заказе */
    $orderinfo = [];
    /* Сохранение информации о конкретном пользователе */
    $orderinfo["name"] = $name;
    $orderinfo["email"] = $email;
    $orderinfo["phone"] = $phone;
    $orderinfo["address"] = $address;
    $orderinfo["orderid"] = $orderid;
    $orderinfo["date"] = $date;
    /* SQL-запрос на выборку из таблицы orders всех товаров для конкретного
покупателя */
    $sql = "SELECT title, author, pubyear, price, quantity
            FROM orders
            WHERE orderid = '$orderid' AND datetime = $date";
    /* Получение результата выборки */
    if(!$result = mysqli_query($link, $sql))
        return false;
    $items = mysqli_fetch_all($result, MYSQLI_ASSOC);
    mysqli_free_result($result);
    /* Сохранение результата в промежуточном массиве */
    $orderinfo["goods"] = $items;
    /* Добавление промежуточного массива в возвращаемый массив */
    $allorders[] = $orderinfo;
}
return $allorders;
```

- Сохраните файл **eshop\inc\lib.inc.php**

Задание 2: Формирование списка заказов

- В текстовом редакторе откройте файл **eshop\admin\orders.php**

- Вызовите функцию **getOrders** и сохраните возвращаемые ей данные в переменную **\$orders**
- Используя цикл как **foreach(\$orders as \$order)** заполните персональные данные покупателей примерно так:

```
<p><b>Заказчик</b>: <?= $order["name"] ?></p>
```
- Выведите дату и время заказа в формате **dd-mm-yyyy hh:mm**
- После заполнения персональных данных, заполните таблицу по аналогии с заполнением корзины (Практическая работа. Часть 6, Задание 2):
- Выведите общую сумму заказа
- Сохраните файл **eshop\admin\orders.php**
- Откройте браузер и в адресной строке наберите: **http://localhost/eshop/admin**
- Перейдите по ссылке Просмотр готовых заказов. Вы должны попасть на страницу **orders.php**
- Убедитесь в корректности вывода списка заказов. В случае ошибок, найдите их и исправьте

Практическая работа. Часть 10

Повышение безопасности приложения

Содержание работы. Часть 10

Повышение безопасности приложения

Задание 1: Создание файла пользователей

- В текстовом редакторе откройте файл **eshop\admin\secure\secure.inc.php**
- Создайте константу для хранения паролей пользователей **FILE_NAME** со значением **.htpasswd**
- Создайте и опишите функцию **getHash**, генерирующую хеш пароля:

```
function getHash($password){  
    $hash = password_hash($password, PASSWORD_BCRYPT);  
    return $hash;  
}
```
- Создайте и опишите функцию **checkHash**, проверяющую пароль:

```
function checkHash($password, $hash){  
    return password_verify($password, $hash);  
}
```
- Создайте и опишите функцию **saveUser**, создающую новую запись в файле пользователей:

```
function saveUser($login, $hash){  
    $str = "$login:$hash\n";  
    if(file_put_contents(FILE_NAME, $str, FILE_APPEND))  
        return true;  
    else  
        return false;  
}
```
- Создайте и опишите функцию **userExists**, проверяющую наличие пользователя в списке:

```
function userExists($login){  
    if(!is_file(FILE_NAME))  
        return false;  
    $users = file(FILE_NAME);  
    foreach($users as $user){  
        if(strpos($user, $login.':') !== false)  
            return $user;  
    }  
    return false;  
}
```
- Сохраните файл **eshop\admin\secure\secure.inc.php**
- В текстовом редакторе откройте файл **eshop\admin\secure\create_user.php**
- Изучите, каким образом происходит создание нового пользователя
- Откройте браузер и в адресной строке введите: **http://localhost/eshop/admin**
- Кликните по ссылке **Добавить пользователя**
- На странице добавления нового пользователя заполните все необходимые поля и нажмите

кнопку **Создать**

- Убедитесь, что файл **.htpasswd** создается и данные пользователя записаны корректно
- Попробуйте создать другого пользователя
- Попробуйте отправить форму не заполняя некоторые поля
- Попробуйте создать пользователя с уже существующим именем
- В случае ошибок, найдите их и исправьте

Задание 2: Включение защиты

- В текстовом редакторе откройте файл **eshop\admin\secure\login.php**
- В начале файла допишите код:

```
session_start();
header("HTTP/1.0 401 Unauthorized");
require_once "secure.inc.php";
if($_SERVER['REQUEST_METHOD']=='POST'){
    $login = trim(strip_tags($_POST["login"]));
    $pw = trim(strip_tags($_POST["pw"]));
    $ref = trim(strip_tags($_GET["ref"]));
    if(!$ref)
        $ref = '/eshop/admin/';
    if($login and $pw){
        if($result = userExists($login)){
            list($_, $hash) = explode(':', $result);
            if(checkHash($pw, $hash)){
                $_SESSION['admin'] = true;
                header("Location: $ref");
                exit;
            }else{
                $title = 'Неправильное имя пользователя или пароль!';
            }
        }else{
            $title = 'Неправильное имя пользователя или пароль!';
        }
    }else{
        $title = 'Заполните все поля формы!';
    }
}
```
- Сохраните файл **eshop\admin\secure\login.php**
- В текстовом редакторе откройте файл **eshop\admin\secure\session.inc.php**
- Введите следующий текст:

```
session_start();
if(!isset($_SESSION['admin'])){
    header('Location: /eshop/admin/secure/login.php?ref=' .
$_SERVER['REQUEST_URI']);
    exit;
}
```

- Сохраните файл **eshop\admin\secure\session.inc.php**
- В текстовом редакторе откройте файл **eshop\admin\secure\secure.inc.php**
- Создайте и опишите функцию **logOut()**, завершающую сеанс пользователя:

```
function logOut(){
    session_destroy();
    header('Location: secure/login.php');
    exit;
}
```
- Сохраните файл **eshop\admin\secure\secure.inc.php**
- В текстовом редакторе откройте файл **eshop\admin\index.php**
- В начале файла допишите код добавив условие:

```
if(isset($_GET['logout'])){
    logOut();
}
```
- Сохраните файл **eshop\admin\index.php**
- Откройте браузер и в адресной строке введите: **http://localhost/eshop/admin**
- Протестируйте защиту. В случае ошибок, найдите их и исправьте

Что нужно ещё сделать?

- Добавить возможность изменять количество товара в корзине с возможностью пересчёта общей цены
- Добавить карточку товара
- Добавить ссылку из каталога на карточку товара
- Добавить возможность загружать картинку товара
- Добавить возможность выставления рейтинга товара
- Добавить комментарии к товару
- Добавить возможность регистрации пользователей
- ... совершенству нет предела

Что почитать?

- Документация РНР

Что дальше?

- РНР. Уровень 3. Профессиональная разработка на РНР