

Лекция 5. Работа с контентом (БД).

Организация поиска

1. Работа с контентом (лекции, лабораторные работы)

Основной контент (материалы лекций, лабораторных работ) предполагается размещать в БД в двух таблицах: lectures, labworks (ранее были продемонстрированы). Учитывая, что за основной вариант формирования меню выберем вариант с полным хранением всех пунктов меню в БД, то необходимо несколько изменить таблицы lectures и labworks – больше нет необходимости хранить в них наименование пунктов меню второго уровня, но для синхронизации и обеспечения целостности данных необходимо организовать связи между таблице й menu и таблицами lectures и labworks соответственно. Для этого в таблицах lectures и labworks добавим поле id (в обеих будет называться одинаково, что создаст определенные удобства при разработке кода).

Таблица *menu*

Сервер: 127.0.0.1:3306 » База данных: my_site » Таблица: menu

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1	id	int(10)		UNSIGNED	Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/>	2	title	varchar(32)	utf8_general_ci		Нет	Нет		
<input type="checkbox"/>	3	parent_id	int(10)		UNSIGNED	Нет	Нет		

↑ ☐ Отметить все С отмеченными:

Сервер: 127.0.0.1:3306 » База данных: my_site » Таблица: menu

Обзор Структура SQL Поиск Вставить

Отображение строк 0 - 7 (8 всего, Запрос занял 0,0000 сек.)

SELECT * FROM `menu`

☐ Профилировать

☐ Показать все | Количество строк: 50 | Фильтровать с

+ Параметры

	id	title	parent_id
<input type="checkbox"/>	1	Лекции	0
<input type="checkbox"/>	2	Лабораторные работы	0
<input type="checkbox"/>	3	Самостоятельная работа	0
<input type="checkbox"/>	7	Лекция 1	1
<input type="checkbox"/>	8	Лекция 2	1
<input type="checkbox"/>	9	Лекция 3	1
<input type="checkbox"/>	10	Лабораторная работа 1	2
<input type="checkbox"/>	11	Лабораторная работа 2	2

↑ ☐ Отметить все С отмеченными:

Таблица *lectures*

Сервер: 127.0.0.1:3306 » База данных: my_site » Таблица: lectures

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Отображение строк 0 - 2 (3 всего, Запрос занял 0,0000 сек.)

```
SELECT * FROM `lectures`
```

Показать все | Количество строк: 50 | Фильтровать строки: Поиск в таблице | Сортировать по индексу: Ниодного

+ Параметры

	lecture_id	id	topic	name	content
<input type="checkbox"/>	1	7	Основы PHP	Типы Данных в PHP	Четыре скалярных типа: • boolean (двоичные данные...
<input type="checkbox"/>	2	8	Основы PHP	Переменные в PHP	Как и в любом другом языке программирования, в PHP...
<input type="checkbox"/>	3	9	Операции с переменными в PHP	Математические операции	Математические операции

☐ Отметить все С отмеченными:

Сервер: 127.0.0.1:3306 » База данных: my_site » Таблица: lectures

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/> 1	lecture_id	int(10)		UNSIGNED	Нет	Нет		AUTO_INCREMENT	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> 2	id	int(10)		UNSIGNED	Нет	Нет			<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> 3	topic	varchar(256)	cp1251_general_ci		Нет	Нет			<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> 4	name	varchar(256)	cp1251_general_ci		Нет	Нет			<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> 5	content	text	cp1251_general_ci		Нет	Нет			<input type="checkbox"/> <input type="checkbox"/>

☐ Отметить все С отмеченными:

Таблица *labworks*

Сервер: 127.0.0.1:3306 » База данных: my_site » Таблица: labworks

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Отображение строк 0 - 1 (2 всего, Запрос занял 0,0000 сек.)

```
SELECT * FROM `labworks`
```

Показать все | Количество строк: 50 | Фильтровать строки: Поиск в таблице | Сортировать по индексу: Ниодного

+ Параметры

	labwork_id	id	lecture_id	name	content
<input type="checkbox"/>	1	10	1	Работа с переменными	Как и в любом другом языке программирования, в PHP...
<input type="checkbox"/>	2	11	2	Работа с символьными переменными	Если строка определяется в двойных кавычках, либо ...

☐ Отметить все С отмеченными:

Сервер: 127.0.0.1:3306 » База данных: my_site » Таблица: labworks

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Триггеры

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/> 1	labwork_id	int(10)		UNSIGNED	Нет	Нет		AUTO_INCREMENT	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> 2	id	int(10)		UNSIGNED	Нет	Нет			<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> 3	lecture_id	int(11)			Нет	Нет			<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> 4	name	varchar(256)	cp1251_general_ci		Нет	Нет			<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> 5	content	text	cp1251_general_ci		Нет	Нет			<input type="checkbox"/> <input type="checkbox"/>

☐ Отметить все С отмеченными:

Рассмотрим три варианта вывода контента (материалов лекций, лабораторных работ) на страницу:

- с полной перезагрузкой страницы;
- с перезагрузкой плавающего фрейма;
- с использованием AJAX для перезагрузки блока div.

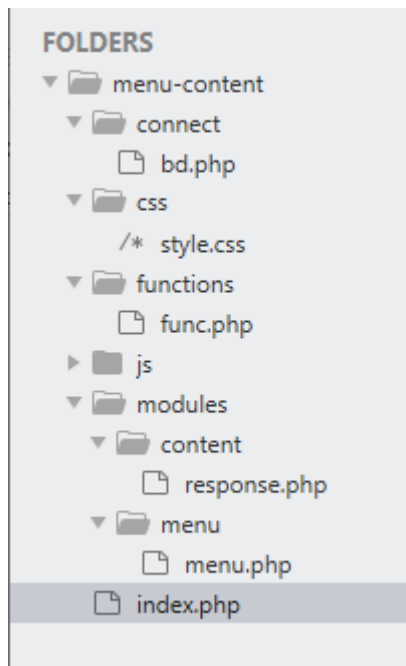
Отметим, что в зависимости от проекта, выбирают чаще либо первый, либо третий вариант реализации данной задачи.

1.1. Вывод контента из БД с полной перезагрузкой страницы

Выполнять вывод контента изначально будем, как уже отмечалось ранее, на небольшом проекте, который будет содержать 2 функциональные возможности: организация меню и вывод контента при переходе пользователя по одному из меню. Внешний вид в таком случае будет следующим.

МЕНЮ САЙТА	<p>Тема лекции: Основы PHP</p> <p>Название лекции: Переменные в PHP</p> <p>Материалы лекции: Как и в любом другом языке программирования, в PHP существует такое понятие, как переменная. При программировании на PHP можно не скупиться на объявление новых переменных. Принципы экономии памяти, которые были актуальны несколько лет назад, сегодня в расчет не принимаются. Однако, при хранении в переменных больших объемов памяти, лучше удалять неиспользуемые переменные, используя оператор <code>unset</code>. Вообще, переменная - это область оперативной памяти, доступ к которой осуществляется по имени. Все данные, с которыми работает программа, хранятся в виде переменных (исключение — константа, которая, впрочем, может содержать только число или строку). Такого понятия, как указатель (как в Си), в PHP не существует — при присвоении переменная копируется один-в-один, какую бы сложную структуру она ни имела. Тем не менее, в PHP, начиная с версии 4, существует понятие ссылок — жестких и символических.</p>
<p>Лекции</p> <ul style="list-style-type: none"> Лекция 1 Лекция 2 Лекция 3 <p>Лабораторные работы</p> <ul style="list-style-type: none"> Лабораторная работа 1 Лабораторная работа 2 <p>Самостоятельная работа</p>	

Структура проекта будет следующей.



Функции, которые будут выполнять извлечение и необходимую сортировку, а также вывод меню на экран, будут храниться в виде отдельного файла `func.php` в каталоге `functions`.

Подключение к БД будет традиционно описано также в виде отдельного файла `bd.php`, который предполагается хранить в папке `connect`.

Стартовый файл `index.php` фактически будет задавать разметку, подключать файл с функциями и далее вызывать их в необходимой последовательности. Также в данном файле будет предусмотрена обработка результатов выбора пользователя – будут формироваться требования для извлечения информации из БД и вызываться файл `response.php`. Последний уже непосредственно выполняет процедуры подключения к БД, извлечения и отображения контента.

Далее представим полные листинги следующих файлов: `index.php`, `func.php`, `response.php` (файл `bd.php` не отличается от данного файла представленного в предыдущих лекциях).

Начнем рассматривать данный проект с файла `index.php` – стартовый файл проекта, который задает разметку страницы (по горизонтали размещены два блока `div`). В первый выводится динамически сформированное меню (подключается файл `menu.php`), а во второй – результаты выборки из БД в соответствии с выбором пользователя (подключается файл `response.php`). Для корректной работы файла `response.php` необходимо считать методом GET переданные по гиперссылке параметры `parent_id` и `id`, а также запись в сессию названия таблицы, из которой будет извлекаться контент.

```
if ((isset($_GET['id'])) && (isset($_GET['parent_id'])))
{
    if ($_GET['parent_id']==1) $_SESSION['where']='lectures';
    if ($_GET['parent_id']==2) $_SESSION['where']='labworks';

    $_SESSION['id']=$_GET['id'];
```

Полный листинг файла **`index.php`** представлен ниже.

```
<?php
    if(session_status() != PHP_SESSION_ACTIVE) session_start();
?>

<!DOCTYPE html>
<html lang="ru" >
    <head>
        <meta http-equiv="content-type" content="text/html; charset=utf-8">

        <link rel="stylesheet" href="css/style.css">
        <script src="js/jquery.min.js" type="text/javascript"></script>
    </head>
    <style>
        ul {
            list-style: none;
            margin: 5;
            margin-right: 20;
            padding-left: 10;
            padding-right: 20px;
```

```

    }

    li {
        margin-bottom: 10;
        margin-top: 10;
    }
    button {
        padding: 0;
        border: none;
        font: inherit;
        color: inherit;
        background-color: transparent;
        cursor: pointer;
    }
    a {
        text-decoration: none; /*убираем подчеркивание текста
ССЫЛОК*/
    }

</style>
<body>

    <div class="menu">
        <?php
            include ("modules/menu/menu.php");
        ?>
    </div>
    <div class="content">
        <?php
            if ((isset($_GET['id']))&&(isset($_GET['parent_id'])))
            {
                if ($_GET['parent_id']==1)
$_SESSION['where']='lectures';
                if ($_GET['parent_id']==2)
$_SESSION['where']='labworks';

                $_SESSION['id']=$_GET['id'];
                include 'modules/content/response.php';
            }
        ?>
    </div>
</body>
</html>

```

```
1 <?php
2     if(session_status()!=PHP_SESSION_ACTIVE) session_start();
3 ?>
4
5 <!DOCTYPE html>
6 <html lang="ru" >
7     <head>
8         <meta http-equiv="content-type" content="text/html; charset=utf-8">
9         <link rel="stylesheet" href="css/style.css">
10        <script src="js/jquery.min.js" type="text/javascript"></script>
11    </head>
12    <style>
13        ul {
14            list-style: none;
15            margin: 5;
16            margin-right: 20;
17            padding-left: 10;
18            padding-right: 20px;
19        }
20
21        li {
22            margin-bottom: 10;
23            margin-top: 10;
24        }
25        button {
26            padding: 0;
27            border: none;
28            font: inherit;
29            color: inherit;
30            background-color: transparent;
31            cursor: pointer;
32        }
33        a {
34            text-decoration: none; /*убираем подчеркивание текста ссылок*/
35        }
36    </style>
37    <body>
38
39        <div class="menu">
40            <?php
41                include ("modules/menu/menu.php");
42            ?>
43        </div>
44        <div class="content">
45            <?php
46                if ((isset($_GET['id']))&&(isset($_GET['parent_id'])))
47                {
48                    if ($_GET['parent_id']==1) $_SESSION['where']='lectures';
49                    if ($_GET['parent_id']==2) $_SESSION['where']='labworks';
50
51                    $_SESSION['id']=$_GET['id'];
52                    include 'modules/content/response.php';
53                }
54            ?>
55        </div>
56    </body>
57 </html>
```

Отметим, что меню будет формироваться файлом menu.php на основе информации, хранящейся в таблице menu в БД.

Листинг файла **menu.php** фактически остается неизменным по сравнению со лекцией 2.

```
<html>
<head>
    <meta charset="windows-1251">
    <link rel="stylesheet" href="./css/style.css">
</head>
<style>
    ul {
        list-style: none; /*убираем маркеры списка*/
        margin: 5;
        margin-right: 20;
        padding-left: 10;
    }
```

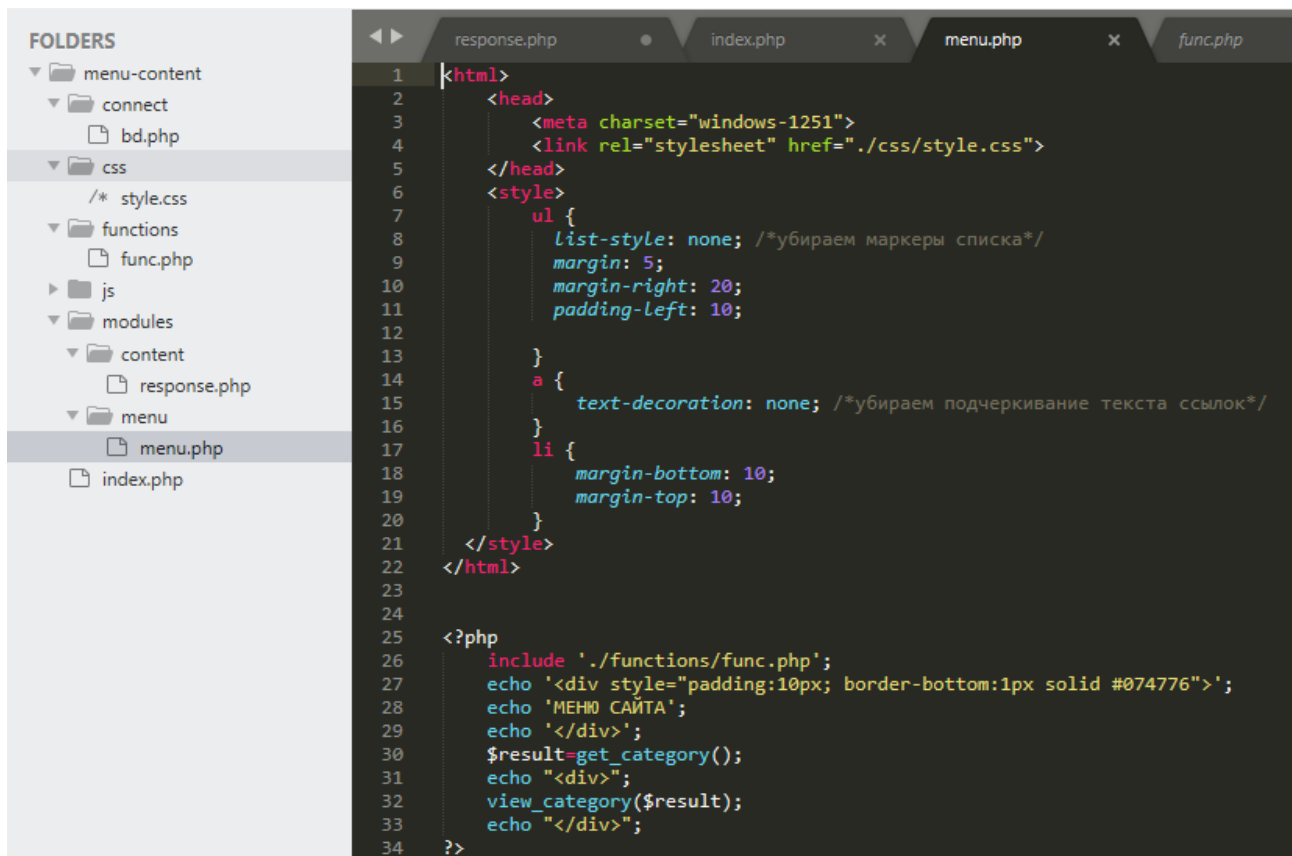
```

        a {
            text-decoration: none; /*убираем подчеркивание текста
ССЫЛОК*/
        }
        li {
            margin-bottom: 10px;
            margin-top: 10px;
        }
    </style>
</html>

<?php
    include './functions/func.php';
    echo '<div style="padding:10px; border-bottom:1px solid #074776">';
    echo 'МЕНЮ САЙТА';
    echo '</div>';
    $result=get_category();
    echo "<div>";
    view_category($result);
    echo "</div>";
?>

```

Переходы по клику пользователя реализуются через гиперссылки. При этом важным является, что методом GET при переходе по гиперссылке дополнительно передаются два параметра: `parent_id` и `id`. Первый из них (`parent_id`) фактически определяет родительскую категорию меню – по ней же будут определяться, из какой таблицы (в данном случае `lectures` или `labworks`) будет извлекаться соответствующая информация. Второй ж



е параметр (id) будет определять id записи в таблице lectures или labworks, которая соответствует искомому контенту.

```

echo '<a
href="index.php?id='.$sarr[$parent_id][$i]['id'].'&parent_id='.$sarr[$parent_id][$i]['parent_id'].'
"target="">'.$sarr[$parent_id][$i]['title'].'</a>';

```

Полный листинг файла **func.php** будет следующим

```

<?php
function get_category()
{
    include './connect/bd.php';
    //запрос к базе данных
    $query ="SELECT * FROM menu";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    if($result)
    {
        $arr_cat = array();
        if(mysqli_num_rows($result) != 0)
        {
            for($i = 0; $i < mysqli_num_rows($result); $i++)
            {
                $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
                if(empty($arr_cat[$row['parent_id']])) {
                    $arr_cat[$row['parent_id']] = array();
                }
                $arr_cat[$row['parent_id']][] = $row;
            }
        }
        return $arr_cat;
    }
}

```



```

    }
}
//вывод каталога с помощью рекурсии
function view_category($arr,$parent_id = 0)
{
    if(empty($arr[$parent_id])) {
        return;
    }
    echo '<ul>';
    for($i = 0; $i < count($arr[$parent_id]);$i++) {
        echo '<li>';
        echo '<a
href="index.php?id='.$arr[$parent_id][$i]['id'].'&parent_id='.$parent_id.
'"target="">'.$arr[$parent_id][$i]['title'].'</a>';
        view_category($arr,$arr[$parent_id][$i]['id']);
        echo '</li>';
    }
    echo '</ul>';
}
?>

```

The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'menu-content', 'connect', 'css', 'functions', 'js', 'modules', and 'content'. The code editor shows the file 'response.php' with the following code:

```

1 <?php
2 function get_category()
3 {
4     include './connect/bd.php';
5     //запрос к базе данных
6     $query = "SELECT * FROM menu";
7     $result = mysqli_query($link, $query) or die("Ошибка " . mysqli_error($
8         link));
9     if($result)
10    {
11        $arr_cat = array();
12        if(mysqli_num_rows($result) != 0)
13        {
14            for($i = 0; $i < mysqli_num_rows($result);$i++)
15            {
16                $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
17                if(empty($arr_cat[$row['parent_id']])) {
18                    $arr_cat[$row['parent_id']] = array();
19                }
20                $arr_cat[$row['parent_id']][] = $row;
21            }
22        }
23        return $arr_cat;
24    }
25 }
26
27 //вывод каталога с помощью рекурсии
28 function view_category($arr,$parent_id = 0)
29 {
30     if(empty($arr[$parent_id])) {
31         return;
32     }
33     echo '<ul>';
34     for($i = 0; $i < count($arr[$parent_id]);$i++) {
35         echo '<li>';
36         echo '<a href="index.php?id='.$arr[$parent_id][$i]['id'].'&parent_id='
37             . $parent_id.'"target="">'.$arr[$parent_id][$i]['title'].'</a>';
38         view_category($arr,$arr[$parent_id][$i]['id']);
39         echo '</li>';
40     }
41     echo '</ul>';
42 }
43 ?>

```

Файл response.php фактически выполняет подключение к БД, считывает из сессии необходимые для запроса параметры и далее непосредственно формирует, и выполняет запрос на извлечение контента.

```
include './connect/bd.php';
```

```

$table=$_SESSION['where'];
$what=$_SESSION['id'];

if ($table=='lectures'){
    $query = "SELECT * FROM $table WHERE id='$what'";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));

```

Остается лишь только выполнить обработку извлеченной из БД информации и отобразить в нужном виде. Полный листинг файла response.php представлен ниже.

Файл **response.php**

```

<?php
    if(session_status()!=PHP_SESSION_ACTIVE) session_start();
?>
<?php
    include './connect/bd.php';

    $table=$_SESSION['where'];
    $what=$_SESSION['id'];

    if ($table=='lectures'){
        $query = "SELECT * FROM $table WHERE id='$what'";
        $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
        if($result)
        {
            $row = mysqli_fetch_assoc($result);

            echo "<p>";
            echo "<b>";
            echo "Тема лекции \";
            echo "</b>";
            echo $row['topic'];
            echo "\"";
            echo "</p>";

            echo "<p>";
            echo "<b>";
            echo "Название лекции \";
            echo "</b>";
            echo $row['name'];
            echo "\"";
            echo "</p>";

            echo "<p>";
            echo $row['content'];
            echo "</p>";
        }
        mysqli_close($link);
    }

    if ($table=='labworks'){
        $query = "SELECT * FROM $table WHERE id='$what'";
        $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));

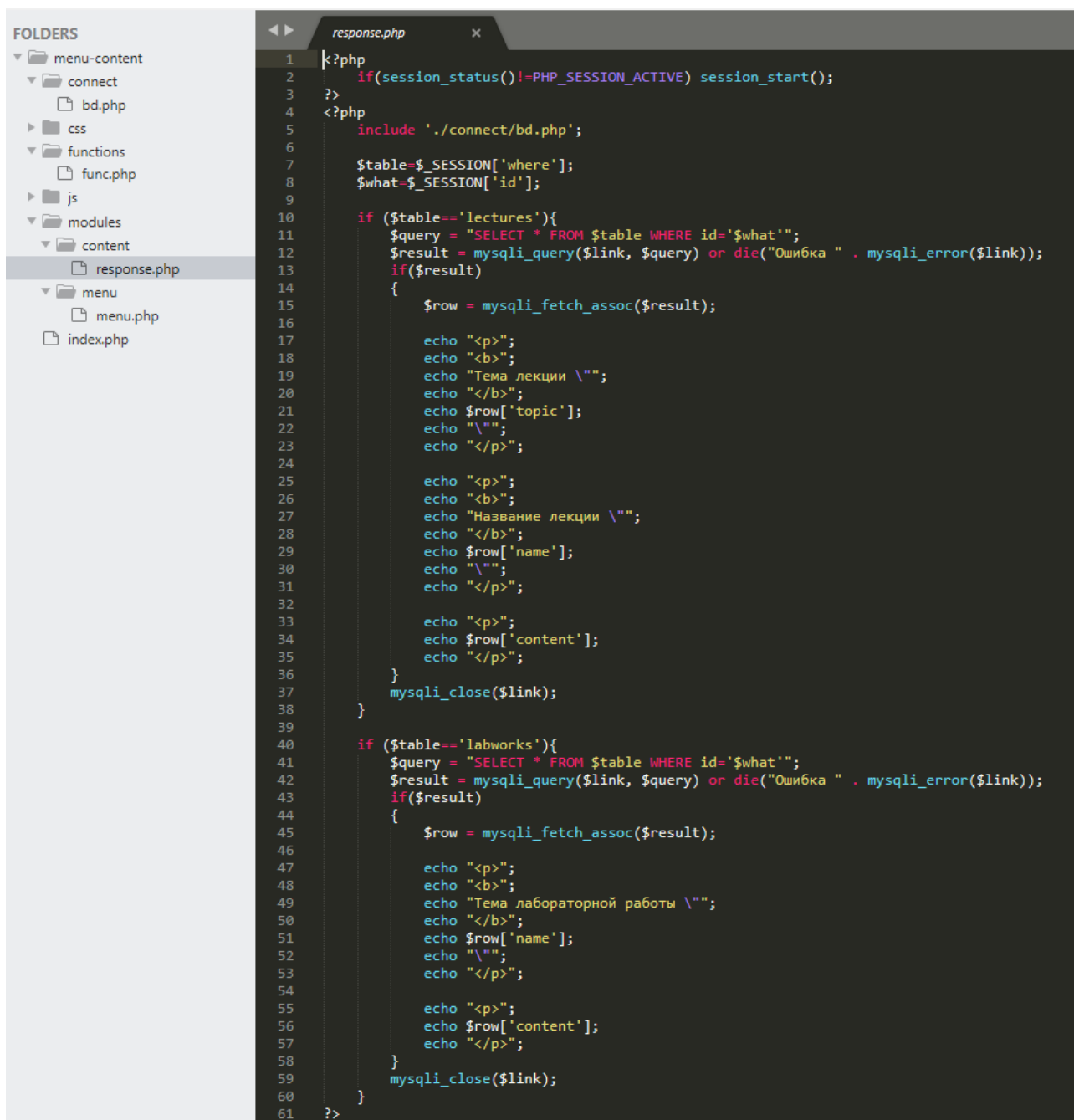
```

```
if($result)
{
    $row = mysqli_fetch_assoc($result);

    echo "<p>";
    echo "<b>";
    echo "Тема лабораторной работы \";
    echo "</b>";
    echo $row['name'];
    echo "\"";
    echo "</p>";

    echo "<p>";
    echo $row['content'];
    echo "</p>";
}
mysqli_close($link);
}
```

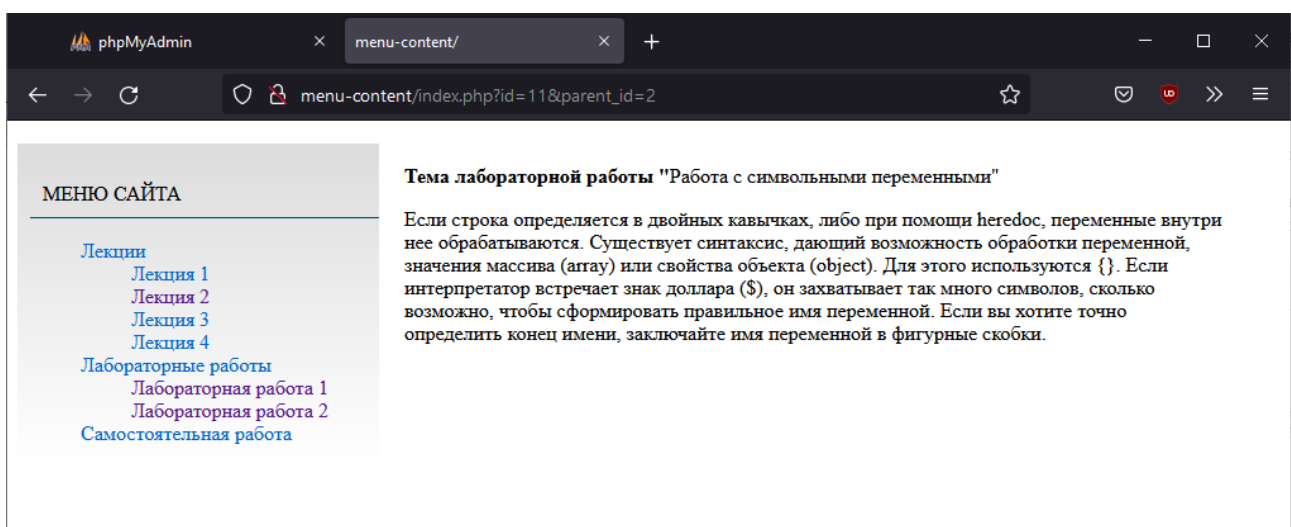
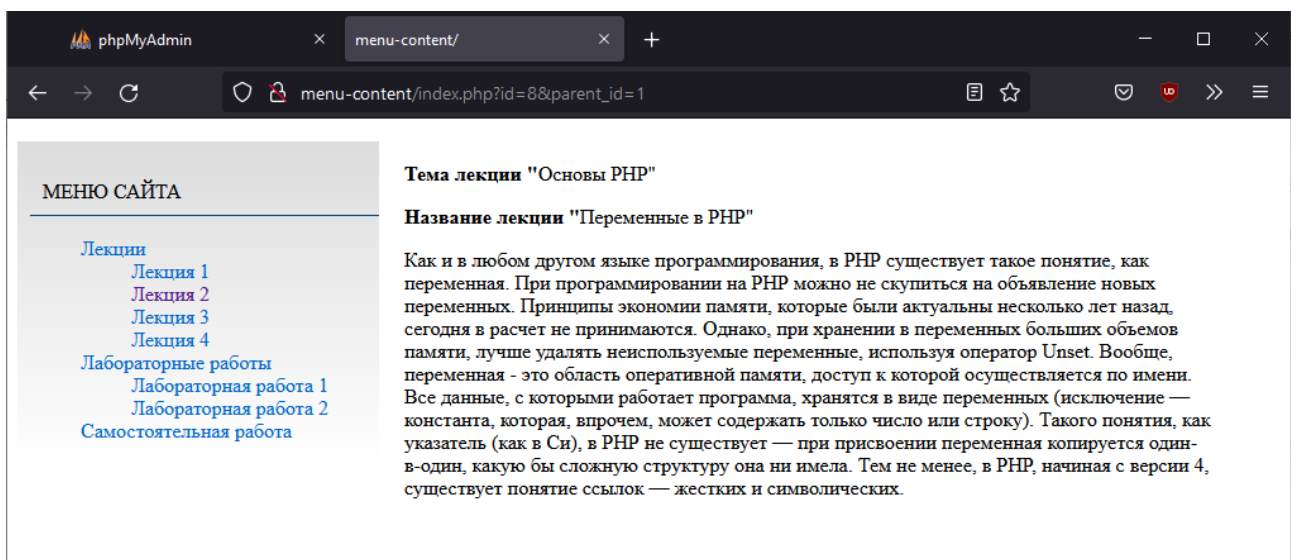
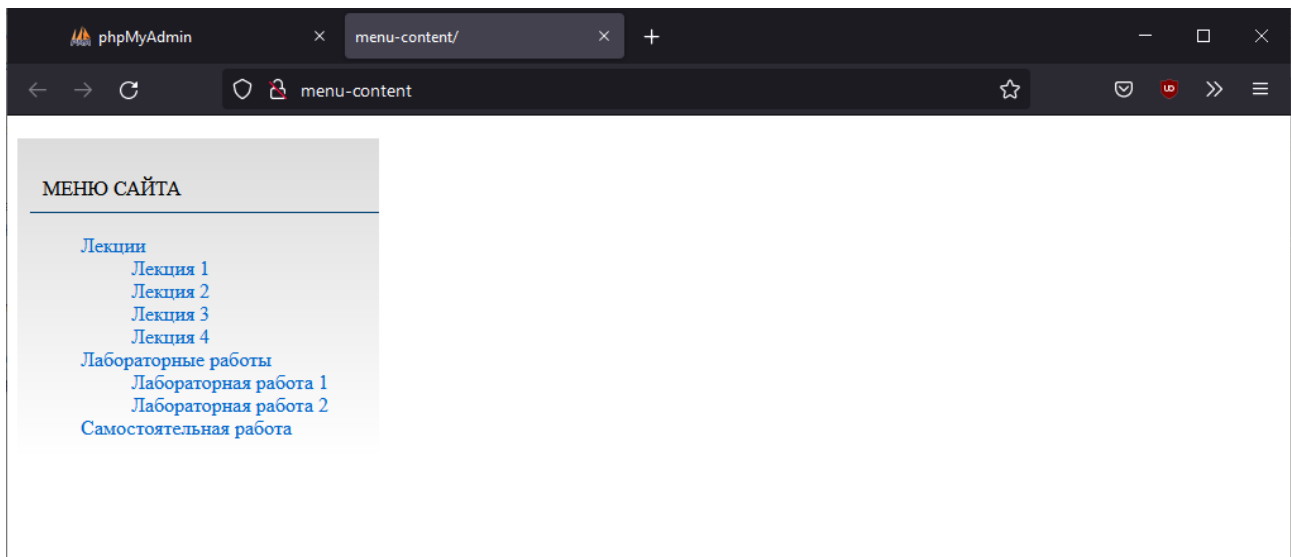
?>



```
1 <?php
2     if(session_status() != PHP_SESSION_ACTIVE) session_start();
3 ?>
4 <?php
5     include './connect/bd.php';
6
7     $table=$_SESSION['where'];
8     $what=$_SESSION['id'];
9
10    if ($table=='lectures'){
11        $query = "SELECT * FROM $table WHERE id='$what'";
12        $result = mysqli_query($link, $query) or die("Ошибка " . mysqli_error($link));
13        if($result)
14        {
15            $row = mysqli_fetch_assoc($result);
16
17            echo "<p>";
18            echo "<b>";
19            echo "Тема лекции \\\\";
20            echo "</b>";
21            echo $row['topic'];
22            echo \\\\";
23            echo "</p>";
24
25            echo "<p>";
26            echo "<b>";
27            echo "Название лекции \\\\";
28            echo "</b>";
29            echo $row['name'];
30            echo \\\\";
31            echo "</p>";
32
33            echo "<p>";
34            echo $row['content'];
35            echo "</p>";
36        }
37        mysqli_close($link);
38    }
39
40    if ($table=='labworks'){
41        $query = "SELECT * FROM $table WHERE id='$what'";
42        $result = mysqli_query($link, $query) or die("Ошибка " . mysqli_error($link));
43        if($result)
44        {
45            $row = mysqli_fetch_assoc($result);
46
47            echo "<p>";
48            echo "<b>";
49            echo "Тема лабораторной работы \\\\";
50            echo "</b>";
51            echo $row['name'];
52            echo \\\\";
53            echo "</p>";
54
55            echo "<p>";
56            echo $row['content'];
57            echo "</p>";
58        }
59        mysqli_close($link);
60    }
61 ?>
```

В итоге получим вывод информации из БД, реализованный через перезагрузку страницы, т.к. при переходе по гиперссылке (из файла `index.php`) вызывается опять же файл `index.php`, но с некоторыми дополнительными параметрами (передаются методом GET и записываются в сессию).

Результаты выполнения проекта представлены на нескольких следующих скриншотах.

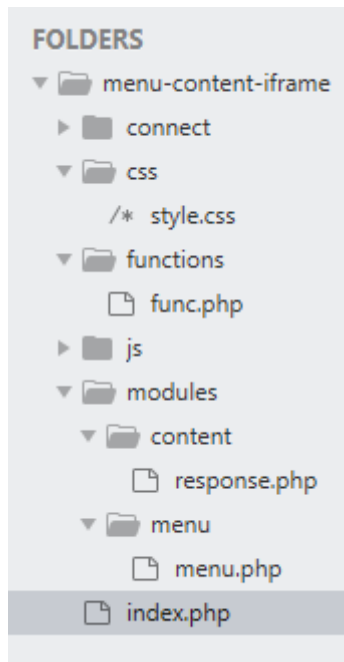


1.2. Вывод контента из БД в iframe

Выполнять вывод контента с использованием `iframe` будем также на небольшом проекте, который также будет содержать 2 функциональные возможности: организация меню и вывод контента при переходе пользователя по одному из меню. Внешний вид фактически будет следующим (совпадает с предыдущим примером).

<div>МЕНЮ САЙТА</div> <div>Лекции</div> <div>Лекция 1</div> <div>Лекция 2</div> <div>Лекция 3</div> <div>Лабораторные работы</div> <div>Лабораторная работа 1</div> <div>Лабораторная работа 2</div> <div>Самостоятельная работа</div>	<p>Тема лекции: Основы PHP</p> <p>Название лекции: Переменные в PHP</p> <p>Материалы лекции: Как и в любом другом языке программирования, в PHP существует такое понятие, как переменная. При программировании на PHP можно не скупиться на объявление новых переменных. Принципы экономии памяти, которые были актуальны несколько лет назад, сегодня в расчет не принимаются. Однако, при хранении в переменных больших объемов памяти, лучше удалять неиспользуемые переменные, используя оператор <code>unset</code>. Вообще, переменная - это область оперативной памяти, доступ к которой осуществляется по имени. Все данные, с которыми работает программа, хранятся в виде переменных (исключение — константа, которая, впрочем, может содержать только число или строку). Такого понятия, как указатель (как в Си), в PHP не существует — при присвоении переменная копируется один-в-один, какую бы сложную структуру она ни имела. Тем не менее, в PHP, начиная с версии 4, существует понятие ссылок — жестких и символических.</p>
--	---

Отметим, что структура проекта при данной стратегии реализации фактически не претерпит изменений. Основные функциональные модули вынесены в отдельные файлы: формирования меню (`//modules/menu/menu.php`) и извлечение/отображение контента (`//modules/content/response.php`). При этом в файле `index.php` в действительности остается только верстка и подключение необходимых файлов.



Функции, которые будут выполнять извлечение и необходимую сортировку, а также вывод меню на экран, будут по-прежнему храниться в виде отдельного файла `func.php` в каталоге `functions`.

Подключение к БД будет традиционно описано также в виде отдельного файла `bd.php`, который предполагается хранить в папке `connect`.

Далее представим полные листинги следующих файлов: index.php, func.php, response.php, menu.php (файл bd.php не отличается от данного файла представленного в предыдущих лекциях).

Файл **index.php** – задает разметку страницы и его полный листинг может быть следующим.

```
<?php
    if(session_status()!=PHP_SESSION_ACTIVE) session_start();
?>

<!DOCTYPE html>
<html lang="ru" >
    <head>
        <meta http-equiv="content-type" content="text/html; charset=utf-
8">
        <link rel="stylesheet" href="css/style.css">
        <script src="js/jquery.min.js" type="text/javascript"></script>
    </head>
    <style>
        ul {
            list-style: none;
            margin: 5;
            margin-right: 20;
            padding-left: 10;
            padding-right: 20px;
        }

        li {
            margin-bottom: 10;
            margin-top: 10;
        }
        button {
            padding: 0;
            border: none;
            font: inherit;
            color: inherit;
            background-color: transparent;
            cursor: pointer;
        }

    </style>
    <body>

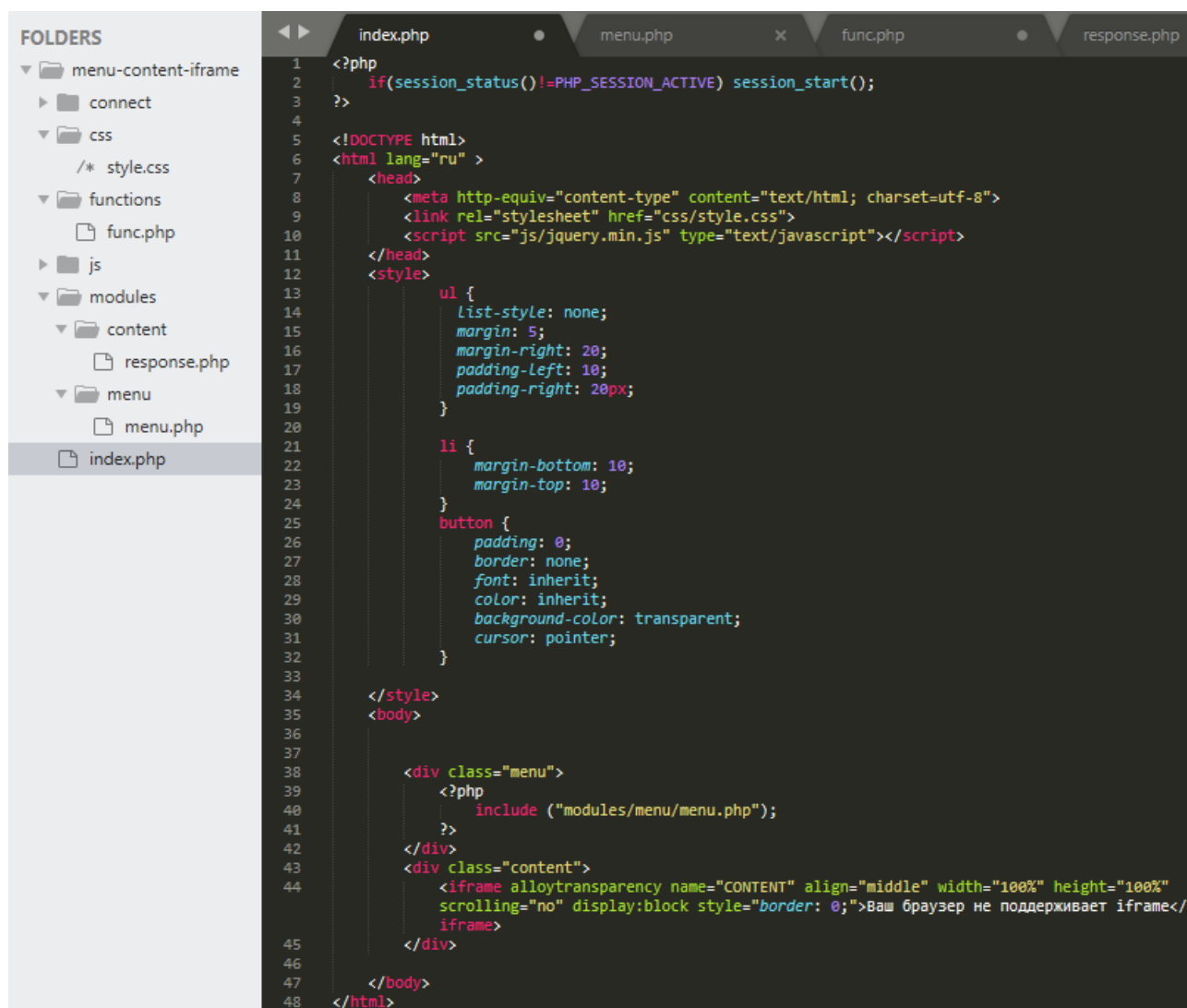
        <div class="menu">
            <?php
                include ("modules/menu/menu.php");
            ?>
        </div>
```

```

        <div class="content">
            <iframe alloytransparency name="CONTENT" align="middle"
width="100%" height="100%" scrolling="no" display:block style="border:
0;">Ваш браузер не поддерживает iframe</iframe>
        </div>

    </body>
</html>

```



Отметим, что меню по-прежнему загружается (подключается файл menu.php) в отдельный блок `<div class="menu">`, а блоку `<div class="content">` имеется плавающий фрейм `iframe`.

Листинг файла **menu.php** фактически не претерпевает изменений.

```

<html>
<head>
    <meta charset="windows-1251">
    <link rel="stylesheet" href="./css/style.css">
</head>
<style>
    ul {
        list-style: none; /*убираем маркеры списка*/
    }

```



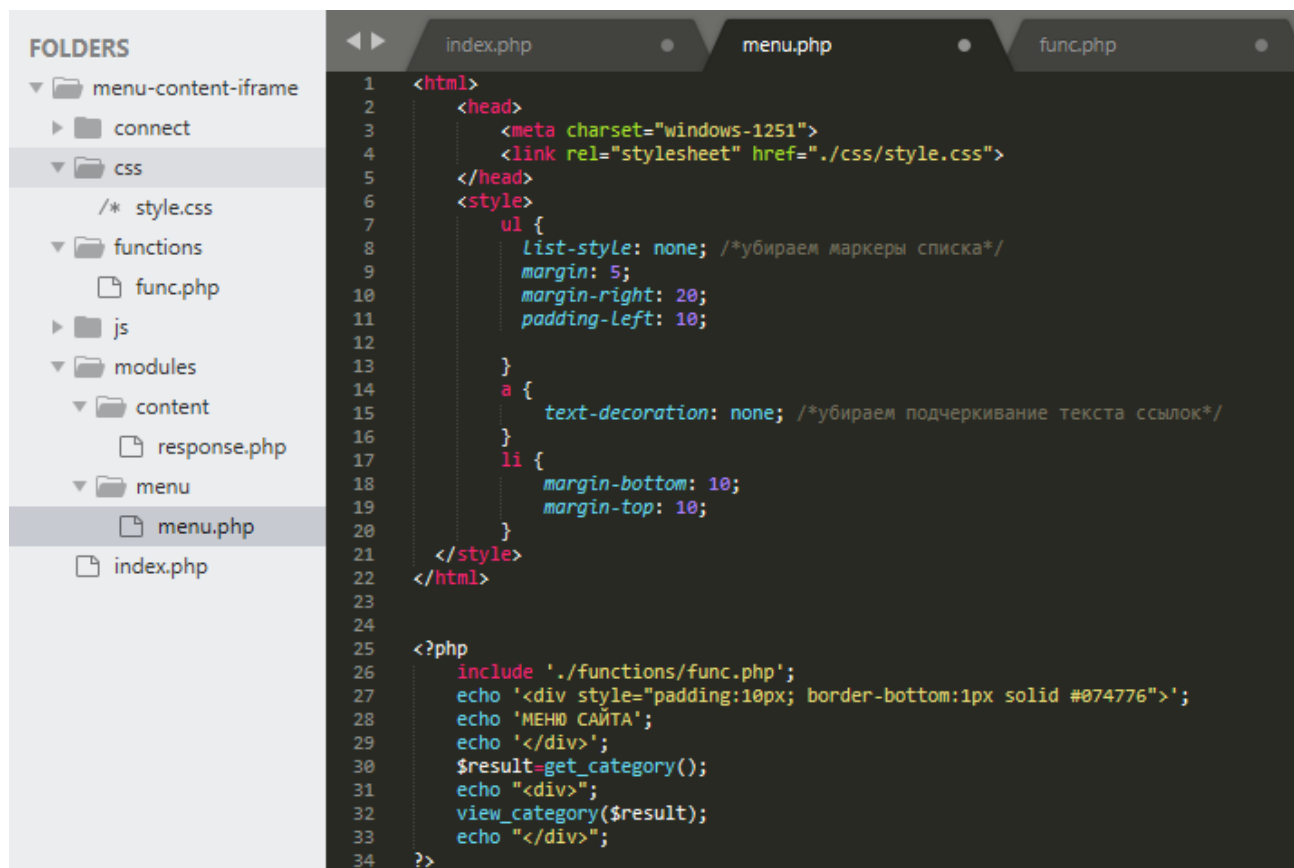
```

        margin: 5;
        margin-right: 20;
        padding-left: 10;

    }
    a {
        text-decoration: none; /*убираем подчеркивание текста
ССЫЛОК*/
    }
    li {
        margin-bottom: 10;
        margin-top: 10;
    }
}
</style>
</html>

<?php
include './functions/func.php';
echo '<div style="padding:10px; border-bottom:1px solid #074776">';
echo 'МЕНЮ САЙТА';
echo '</div>';
$result=get_category();
echo "<div>";
view_category($result);
echo "</div>";
?>

```



Файла **func.php** практически совпадает с версией в предыдущем примере. Единственным отличием является явное указание при формировании гиперссылки имени **iframe**, куда будет выгружаться контент.

```
        echo '<li><a
href="./modules/content/response.php?id='.$sarr[$parent_id][$i]['id'].
        '&parent_id='.$parent_id.'"target="CONTENT">'
        . $sarr[$parent_id][$i]['title'].'</a>';
```

Полный листинг файла **func.php** представлен ниже.

```
<?php
function get_category()
{
    include './connect/bd.php';
    $query = "SELECT * FROM menu";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    if($result)
    {
        $arr_cat = array();
        if(mysqli_num_rows($result) != 0)
        {
            //В цикле формируем массив
            for($i = 0; $i < mysqli_num_rows($result); $i++)
            {
                $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
                if(empty($arr_cat[$row['parent_id']])) {
                    $arr_cat[$row['parent_id']] = array();
                }
                $arr_cat[$row['parent_id']][] = $row;
            }
        }
        //возвращаем массив
        return $arr_cat;
    }
}

//вывод каталога с помощью рекурсии
function view_category($arr, $parent_id = 0)
{
    if(empty($arr[$parent_id])) {
        return;
    }
    echo '<ul>';
    for($i = 0; $i < count($arr[$parent_id]); $i++) {
        echo '<li><a
href="./modules/content/response.php?id='.$arr[$parent_id][$i]['id'].
        '&parent_id='.$parent_id.'"target="CONTENT">'
        . $arr[$parent_id][$i]['title'].'</a>';
```

```

        view_category($arr,$arr[$parent_id][$i]['id']);
    echo '</li>';
}
echo '</ul>';
}
?>

```

```

1 <?php
2 function get_category()
3 {
4     include './connect/bd.php';
5     $query = "SELECT * FROM menu";
6     $result = mysqli_query($link, $query) or die("Ошибка " . mysqli_error($link));
7     if($result)
8     {
9         $arr_cat = array();
10        if(mysqli_num_rows($result) != 0)
11        {
12            //в цикле формируем массив
13            for($i = 0; $i < mysqli_num_rows($result); $i++)
14            {
15                $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
16                if(empty($arr_cat[$row['parent_id']])) {
17                    $arr_cat[$row['parent_id']] = array();
18                }
19                $arr_cat[$row['parent_id']][] = $row;
20            }
21        }
22        //возвращаем массив
23        return $arr_cat;
24    }
25 }
26
27 //вывод каталога с помощью рекурсии
28 function view_category($arr,$parent_id = 0)
29 {
30
31     if(empty($arr[$parent_id])) {
32         return;
33     }
34     echo '<ul>';
35     for($i = 0; $i < count($arr[$parent_id]); $i++) {
36         echo '<li><a href="./modules/content/response.php?id='.$arr[$parent_id][$i]['id'].
37         '&parent_id='.$parent_id.'"target="CONTENT">'
38         . $arr[$parent_id][$i]['title'].'</a>';
39         view_category($arr,$arr[$parent_id][$i]['id']);
40         echo '</li>';
41     }
42     echo '</ul>';
43 }
44 ?>

```

Файл response.php по-прежнему выполняет подключение к БД, считывает (методом GET) необходимые для запроса параметры (parent_id и id, смысловая нагрузка которых аналогична предыдущему примеру) и далее непосредственно формирует, и выполняет запрос на извлечение контента.

```

if ((isset($_GET['id'])) && (isset($_GET['parent_id'])))
{
    if ($_GET['parent_id']==1) $table='lectures';
    if ($_GET['parent_id']==2) $table='labworks';

    $what=$_GET['id'];
}

```

Остается лишь только выполнить обработку извлеченной из БД информации и отобразить в нужно виде. Полный листинг файла **response.php** представлен ниже.

Файл response.php

```
<?php
include '../../connect/bd.php';

if ((isset($_GET['id'])) && (isset($_GET['parent_id'])))
{
    if ($_GET['parent_id']==1) $table='lectures';
    if ($_GET['parent_id']==2) $table='labworks';

    $what=$_GET['id'];
}

if ($table=='lectures'){
    $query = "SELECT * FROM $table WHERE id='$what'";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    if($result)
    {
        $row = mysqli_fetch_assoc($result);

        echo "<p>";
        echo "<b>";
        echo "Тема лекции \";
        echo "</b>";
        echo $row['topic'];
        echo "\"";
        echo "</p>";

        echo "<p>";
        echo "<b>";
        echo "Название лекции \";
        echo "</b>";
        echo $row['name'];
        echo "\"";
        echo "</p>";

        echo "<p>";
        echo $row['content'];
        echo "</p>";
    }
    mysqli_close($link);
}

if ($table=='labworks'){
    $query = "SELECT * FROM $table WHERE id='$what'";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    if($result)
    {
        $row = mysqli_fetch_assoc($result);

        echo "<p>";
        echo "<b>";
        echo "Тема лабораторной работы \";
        echo "</b>";
        echo $row['name'];
        echo "\"";
    }
}
```

```

        echo "</p>";

        echo "<p>";
        echo $row['content'];
        echo "</p>";
    }
    mysqli_close($link);
}

```

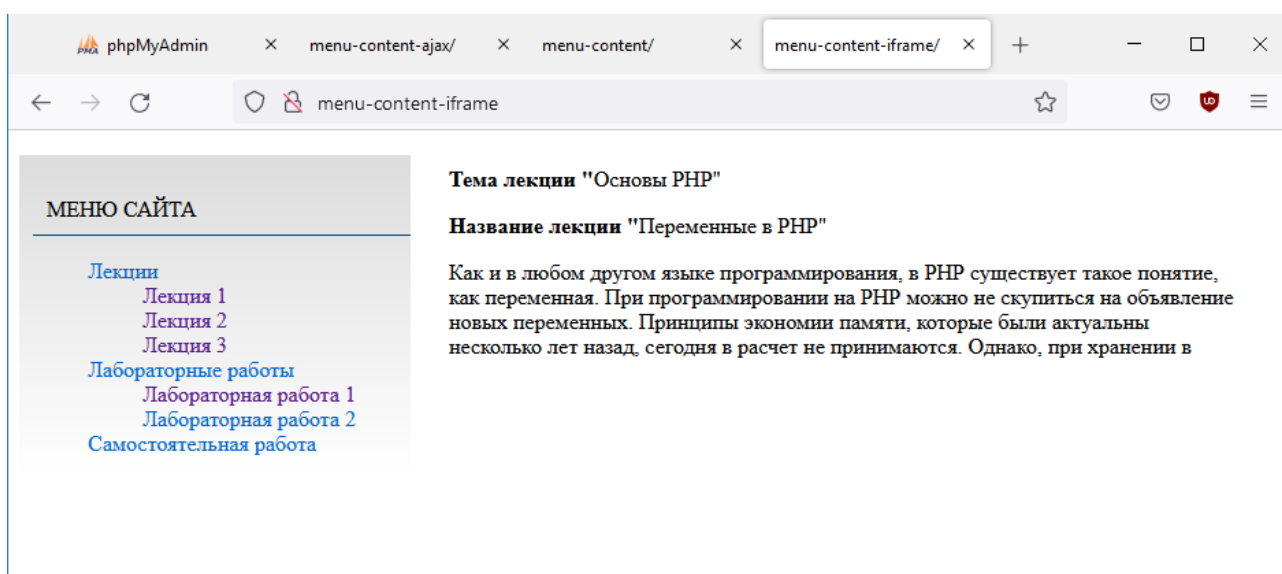
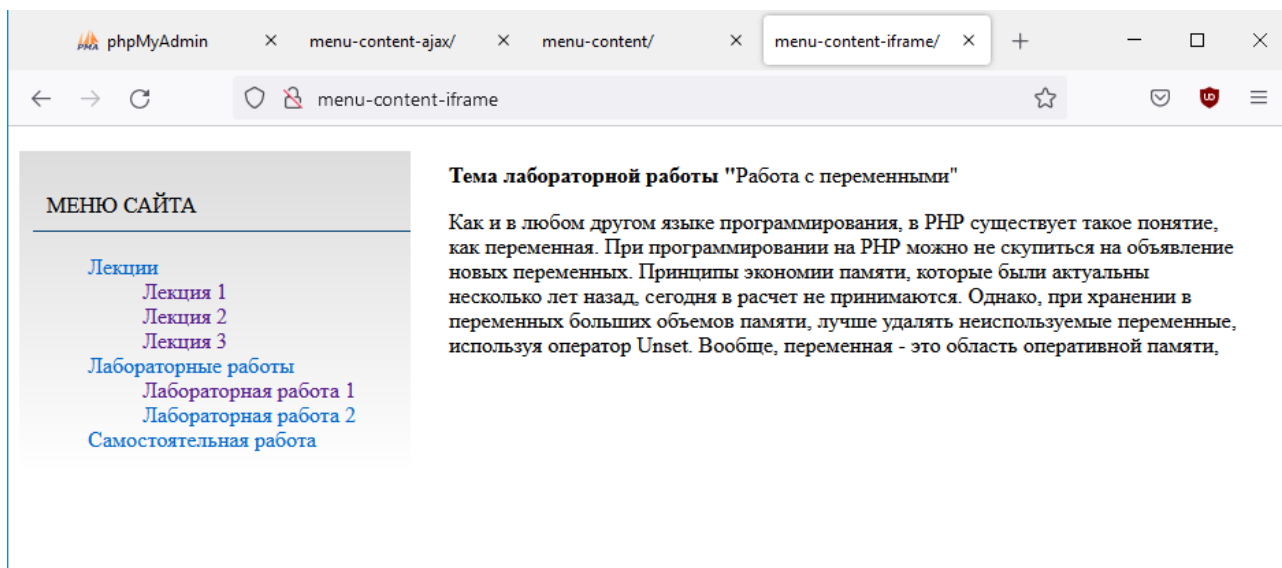
??

```

1  <?php
2  include '../connect/bd.php';
3
4  if ((isset($_GET['id'])) && (isset($_GET['parent_id'])))
5  {
6      if ($_GET['parent_id']==1) $table='lectures';
7      if ($_GET['parent_id']==2) $table='labworks';
8
9      $what=$_GET['id'];
10 }
11
12 if ($table=='lectures'){
13     $query = "SELECT * FROM $table WHERE id='$what'";
14     $result = mysqli_query($link, $query) or die("Ошибка " . mysqli_error($link));
15     if($result)
16     {
17         $row = mysqli_fetch_assoc($result);
18
19         echo "<p>";
20         echo "<b>";
21         echo "Тема лекции \";
22         echo "</b>";
23         echo $row['topic'];
24         echo "\"";
25         echo "</p>";
26
27         echo "<p>";
28         echo "<b>";
29         echo "Название лекции \";
30         echo "</b>";
31         echo $row['name'];
32         echo "\"";
33         echo "</p>";
34
35         echo "<p>";
36         echo $row['content'];
37         echo "</p>";
38     }
39     mysqli_close($link);
40 }
41
42 if ($table=='labworks'){
43     $query = "SELECT * FROM $table WHERE id='$what'";
44     $result = mysqli_query($link, $query) or die("Ошибка " . mysqli_error($link));
45     if($result)
46     {
47         $row = mysqli_fetch_assoc($result);
48
49         echo "<p>";
50         echo "<b>";
51         echo "Тема лабораторной работы \";
52         echo "</b>";
53         echo $row['name'];
54         echo "\"";
55         echo "</p>";
56
57         echo "<p>";
58         echo $row['content'];
59         echo "</p>";
60     }
61     mysqli_close($link);
62 }
63
64 ?>

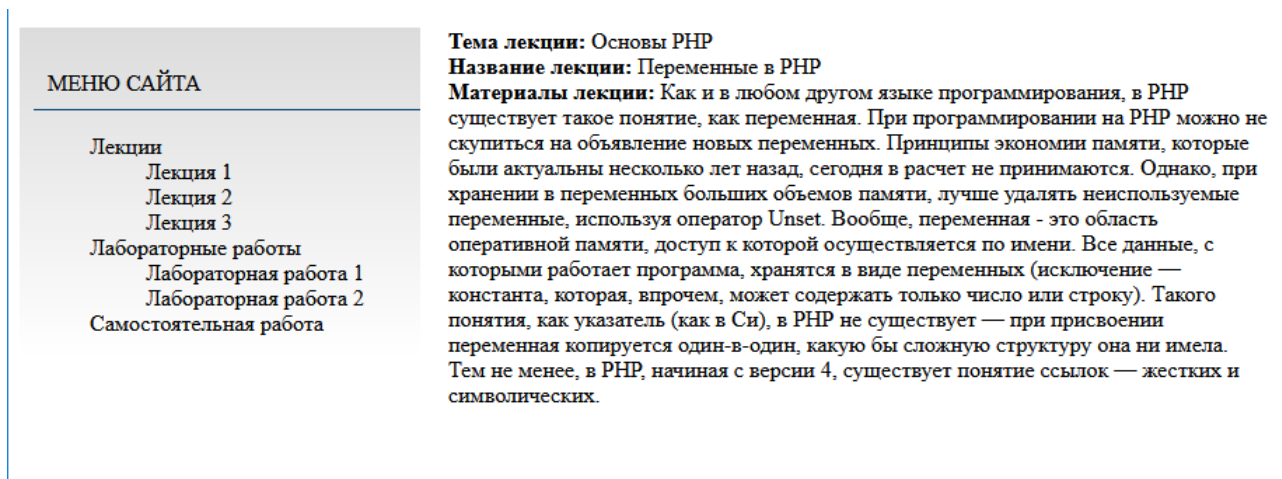
```

Результаты выполнения проекта представлены на нескольких следующих скриншотах.

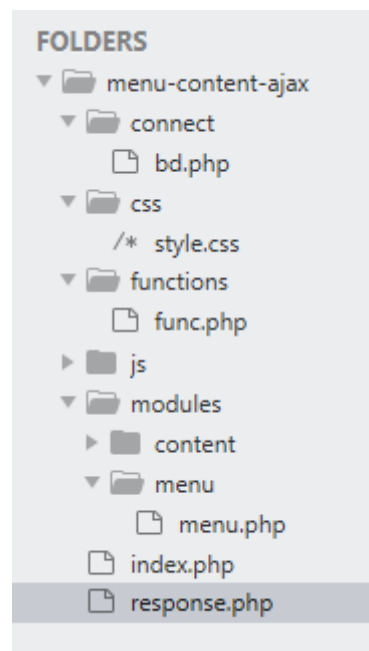


1.3. Вывод контента из БД с перезагрузкой div с использованием AJAX

Выполнять вывод контента с использованием AJAX будем опять же на небольшом проекте, который также будет содержать 2 функциональные возможности: организация меню и вывод контента при переходе пользователя по одному из меню. Внешний вид фактически будет следующим (совпадает с предыдущими вариантами).



Отметим, что структура проекта при данной стратегии реализации фактически не претерпит изменений. Основные функциональные модули вынесены в отдельные файлы: формирования меню (`//modules/menu/menu.php`) и извлечение/отображение контента (`response.php`). При этом в файле `index.php` будет верстка, подключение необходимых файлов, а также JS-скрипты и функции для обработки действий пользователя верстка.



Функции, которые будут выполнять извлечение и необходимую сортировку, а также вывод меню на экран, будут по-прежнему храниться в виде отдельного файла `func.php` в каталоге `functions`.

Подключение к БД будет традиционно описано также в виде отдельного файла `bd.php`, который предполагается хранить в папке `connect`.

Далее рассмотрим коды следующих файлов: `index.php`, `func.php`, `response.php`, `menu.php` (файл `bd.php` не отличается от данного файла представленного в предыдущих лекциях).

Ключевыми логическими моментами в файле **`index.php`** являются:

- JS-скрипт для обработки действия пользователя (клик по пункту меню, реализованного не через гиперссылку, как было ранее, а через кнопку (button)).

```
document.body.addEventListener('click', () => {
    var target = event.target;
    if (target.id==1) content_lectures (target.value);
    if (target.id==2) content_labworks (target.value);
})
```

- Две ajax-функции, предназначенные для асинхронного запроса, связанного с извлечением контента из таблиц `lectures` и `labworks`.

```
function content_lectures (id){
    $.ajax({
        dataType: 'json',
        url: 'response.php?where=lectures&what='+id,
        success: function(jsondata)
        {
            $('<div>.content</div>').html('<b>Тема лекции:</b> ' +
            jsondata.topic + '<br><b>Название лекции:</b> ' + jsondata.name+
            '<br><b>Материалы лекции:</b> ' + jsondata.content);
        }
    })
}
```

```

        });
    };
    function content_labworks (id){
        $.ajax({
            dataType: 'json',
            url: 'response.php?where=labworks&what='+id,
            success: function(jsondata)
            {
                $('#.content').html('<b>Тема лабораторной
работы:</b> ' + jsondata.name + '<br><b>Материалы лабораторной
работы:</b> ' + jsondata.content);
            }
        });
    };
};

```

- Разметка задается привычным способом – на блоках div. В блоке для отображения меню, подключаются соответствующие файлы – func.php и menu.php.

```

<div class="menu">
    <?php
        include './functions/func.php';
        include './modules/menu/menu.php';
    ?>
</div>
<div class="content">

</div>

```

Полный листинг файла **index.php** представлен ниже.

```

<!DOCTYPE html>
<html lang="ru" >
    <head>
        <meta http-equiv="content-type" content="text/html; charset=utf-8">
        <link rel="stylesheet" href="css/style.css">
        <script src="js/jquery.min.js" type="text/javascript"></script>
    </head>
    <style>
        ul {
            list-style: none;
            margin: 5;
            margin-right: 20;
            padding-left: 10;
            padding-right: 20px;
        }

        li {
            margin-bottom: 10;
            margin-top: 10;
        }
        button {
            padding: 0;
            border: none;
            font: inherit;
            color: inherit;

```



```

        background-color: transparent;
        cursor: pointer;
    }
</style>
<body>
    <script language="javascript" type="text/javascript">

        document.body.addEventListener('click', () => {
            var target = event.target;
            if (target.id==1) content_lectures (target.value);
            if (target.id==2) content_labworks (target.value);
        })

        function content_lectures (id){
            $.ajax({
                dataType: 'json',
                url: 'response.php?where=lectures&what='+id,
                success: function(jsondata)
                {
                    $('#.content').html('<b>Тема лекции:</b> ' +
jsondata.topic + '<br><b>Название лекции:</b> ' + jsondata.name+
'<br><b>Материалы лекции:</b> ' + jsondata.content);
                }
            });
        };

        function content_labworks (id){
            $.ajax({
                dataType: 'json',
                url: 'response.php?where=labworks&what='+id,
                success: function(jsondata)
                {
                    $('#.content').html('<b>Тема лабораторной
работы:</b> ' + jsondata.name + '<br><b>Материалы лабораторной
работы:</b> ' + jsondata.content);
                }
            });
        };

    </script>

    <div class="menu">
        <?php
            include './functions/func.php';
            include './modules/menu/menu.php';
        ?>
    </div>
    <div class="content">

        </div>
    </body>
</html>

```

Файл, предназначенный для организации меню **menu.php**, не претерпел изменений, а вот функция `view_category`, предназначенная для отображения пунктов меню, изменилась – связаны они с использованием элементов формы типа «button» вместо гиперссылок.

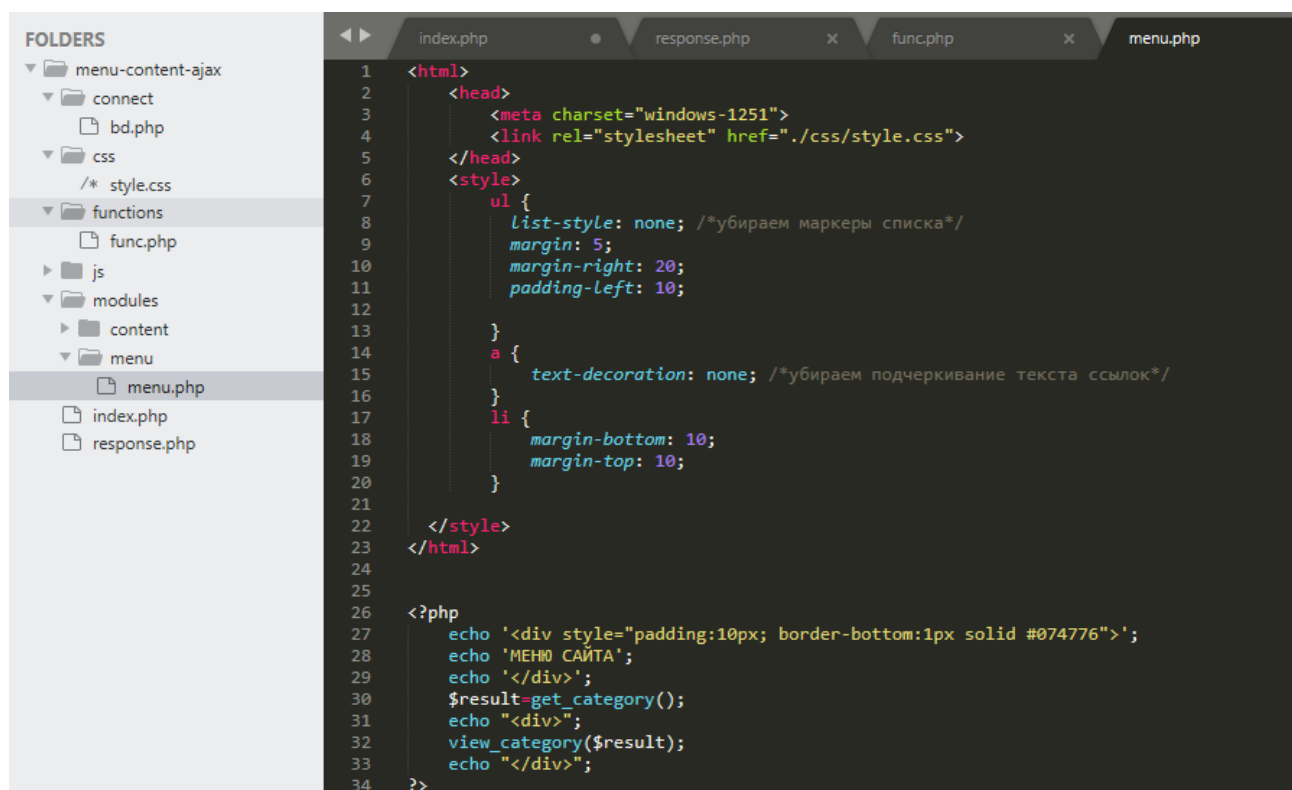
Файл menu.php

```
<html>
  <head>
    <meta charset="windows-1251">
    <link rel="stylesheet" href="./css/style.css">
  </head>
  <style>
    ul {
      list-style: none; /*убираем маркеры списка*/
      margin: 5;
      margin-right: 20;
      padding-left: 10;

    }
    a {
      text-decoration: none;
    }
    li {
      margin-bottom: 10;
      margin-top: 10;
    }
  </style>
</html>

<?php
  echo '<div style="padding:10px; border-bottom:1px solid #074776">';
  echo 'МЕНЮ САЙТА';
  echo '</div>';
  $result=get_category();
  echo "<div>";
  view_category($result);
  echo "</div>";

?>
```



Ключевым изменением в файла **func.php** является следующий блок строк кода.

```
echo '<ul>';
    for($i = 0; $i < count($arr[$parent_id]);$i++) {
        echo '<li>';
        echo "<button value=\".$arr[$parent_id][$i]['id'].\"
id=\".$arr[$parent_id][$i]['parent_id'].\">\".$arr[$parent_id][$i]['title'].
\"</button>\";
        view_category($arr,$arr[$parent_id][$i]['id']);
        echo '</li>';
    }
echo '</ul>';
```

Отметим, что в качестве параметров кнопки используются **value**, которое будет хранить **id** лекции или лабораторной работы, и параметр **id**, который будет хранить **parent_id** родительской категории. Эти два параметра исключительно важны для корректного перехода на материалы лекций и лабораторных работ.

Полный листинг файла **func.php** представлен ниже.

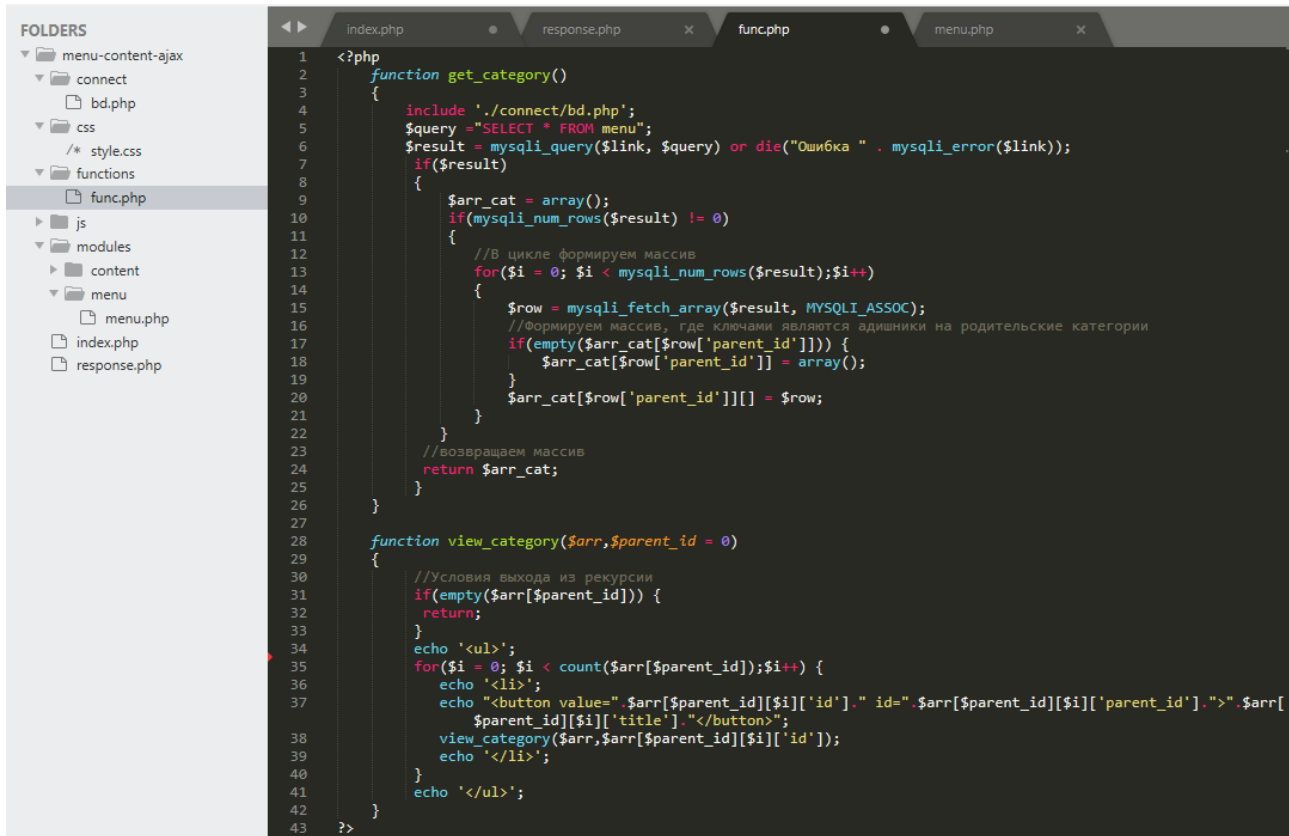
```
<?php
function get_category() {
    include './connect/bd.php';
    $query = "SELECT * FROM menu";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    if($result)
    {
        $arr_cat = array();
        if(mysqli_num_rows($result) != 0)
        {
            //В цикле формируем массив
            for($i = 0; $i < mysqli_num_rows($result);$i++)
            {
                $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
                //Формируем массив, где ключами являются адишники на
родительские категории
                if(empty($arr_cat[$row['parent_id']])) {
                    $arr_cat[$row['parent_id']] = array();
                }
                $arr_cat[$row['parent_id']][] = $row;
            }
        }
        //возвращаем массив
        return $arr_cat;
    }
}

function view_category($arr,$parent_id = 0) {
    //Условия выхода из рекурсии
    if(empty($arr[$parent_id])) {
        return;
    }
    echo '<ul>';
    for($i = 0; $i < count($arr[$parent_id]);$i++) {
        echo '<li>';
        echo "<button value=\".$arr[$parent_id][$i]['id'].\"
id=\".$arr[$parent_id][$i]['parent_id'].\">\".$arr[$parent_id][$i]['title'].
\"</button>\";
    }
}
```

```

        view_category($arr, $arr[$parent_id][$i]['id']);
        echo '</li>';
    }
    echo '</ul>';
}
?>

```



Файл `response.php` будет в таком случае выполнять подключение к БД, считывать параметры sql-запроса (`parent_id` и `id`, смысловая нагрузка которых аналогична предыдущему примеру) и далее непосредственно формировать, и выполняет запрос на извлечение контента. Результат возвращается в аякс-функцию в json-формате.

Полный листинг файла **`response.php`** представлен ниже.

Файл **`response.php`**

```

<?php
    include './connect/bd.php';
    $table=$_REQUEST['where'];
    $what=$_REQUEST['what'];
    $query = "SELECT * FROM $table WHERE id='$what'";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));

    if($result)
    {
        $row = mysqli_fetch_assoc($result);
    }
    mysqli_close($link);
    echo json_encode($row);
?>

```

FOLDERS

menu-content-ajax

connect

bd.php

css

/* style.css

functions

func.php

js

modules

content

menu

menu.php

index.php

response.php

index.php

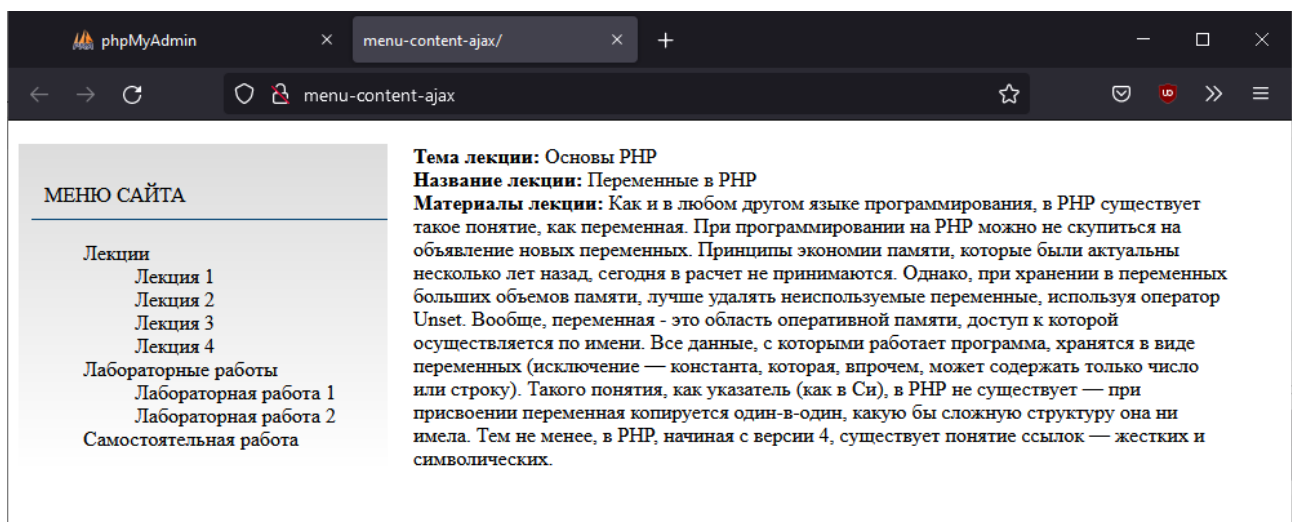
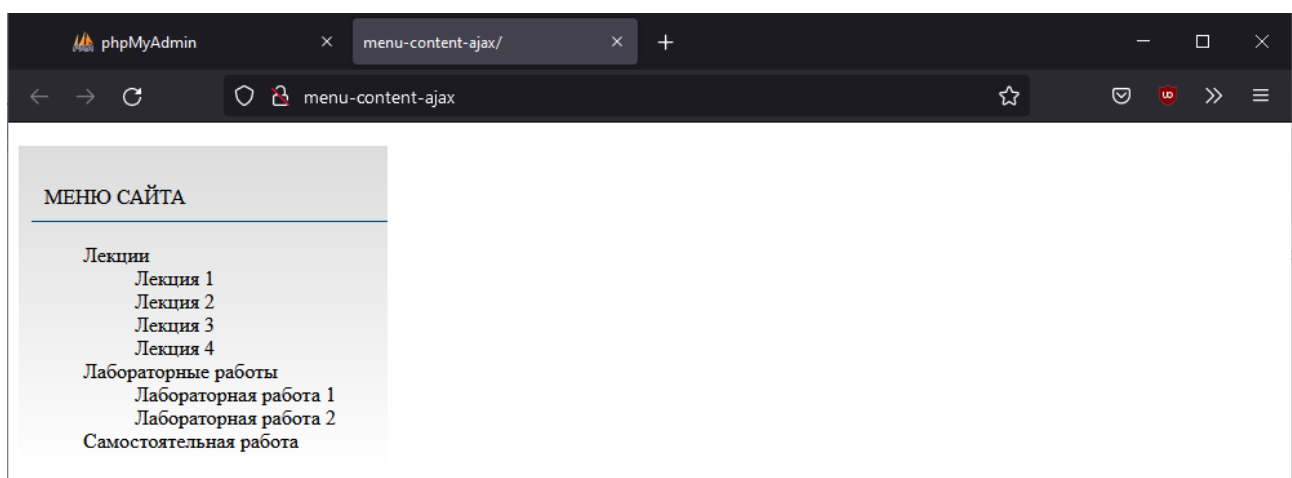
response.php

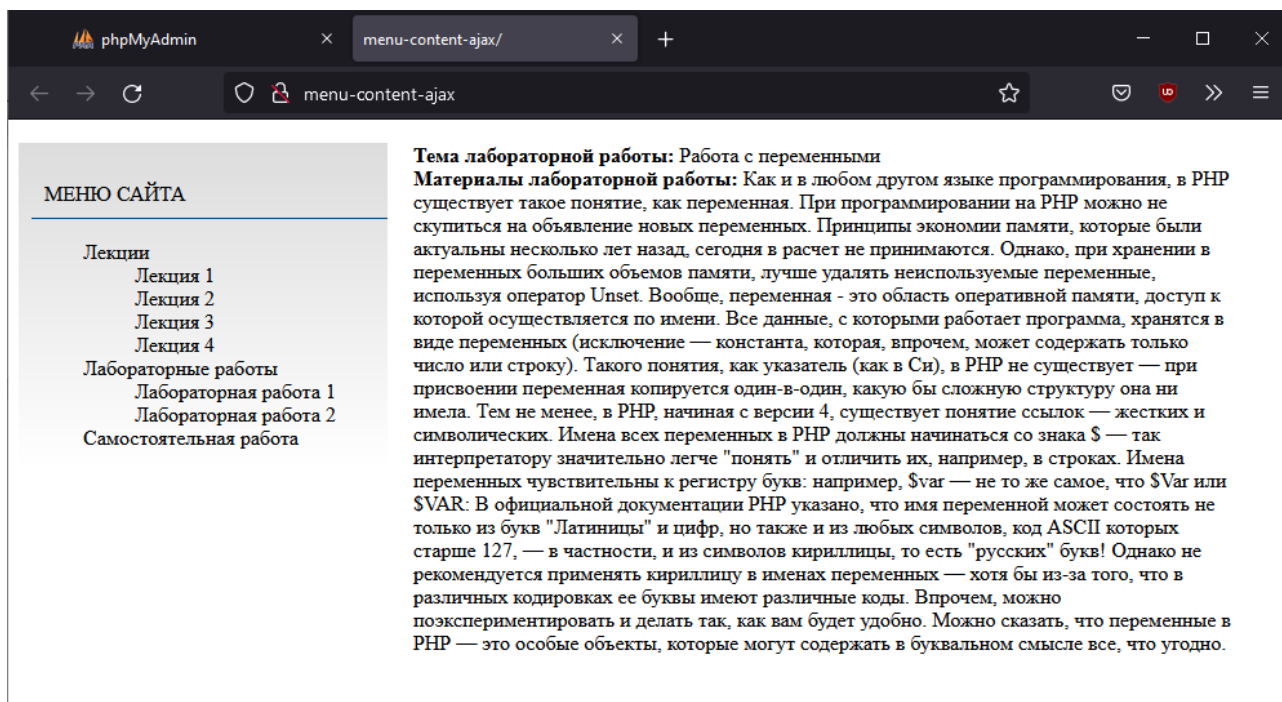
func.php

menu.php

```
1 <?php
2     include './connect/bd.php';
3
4     $table=$_REQUEST['where'];
5     $what=$_REQUEST['what'];
6
7     $query = "SELECT * FROM $table WHERE id='$what'";
8     $result = mysqli_query($link, $query) or die("Ошибка " . mysqli_error($link));
9
10    if($result)
11    {
12        $row = mysqli_fetch_assoc($result);
13    }
14    mysqli_close($link);
15    echo json_encode($row);
16 ?>
17
18
19
```

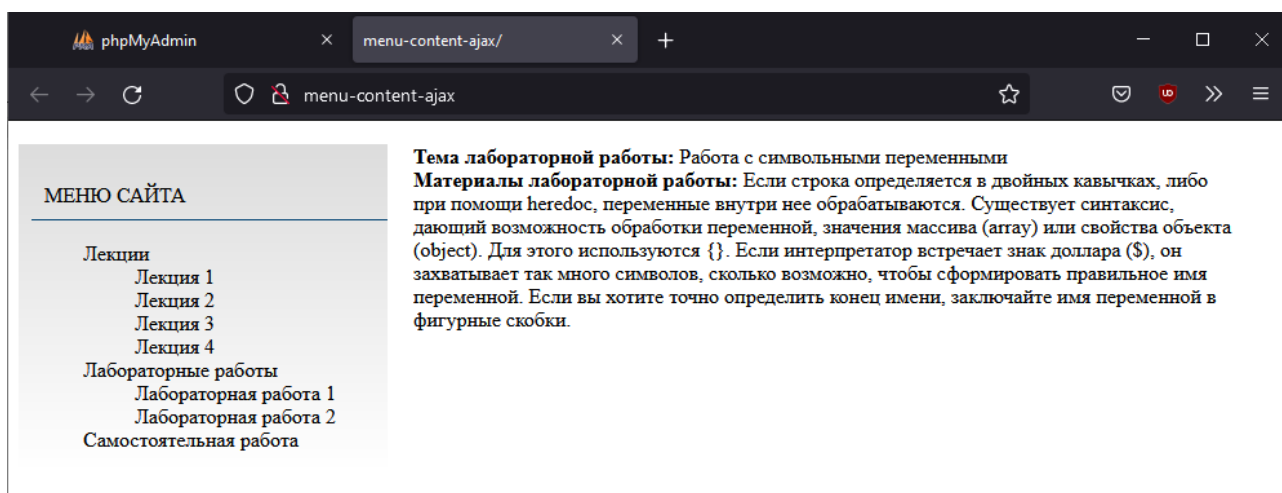
Результаты выполнения проекта представлены на нескольких следующих скриншотах





Тема лабораторной работы: Работа с переменными

Материалы лабораторной работы: Как и в любом другом языке программирования, в PHP существует такое понятие, как переменная. При программировании на PHP можно не скупиться на объявление новых переменных. Принципы экономии памяти, которые были актуальны несколько лет назад, сегодня в расчет не принимаются. Однако, при хранении в переменных больших объемов памяти, лучше удалять неиспользуемые переменные, используя оператор `unset`. Вообще, переменная - это область оперативной памяти, доступ к которой осуществляется по имени. Все данные, с которыми работает программа, хранятся в виде переменных (исключение — константа, которая, впрочем, может содержать только число или строку). Такого понятия, как указатель (как в Си), в PHP не существует — при присвоении переменная копируется один-в-один, какую бы сложную структуру она ни имела. Тем не менее, в PHP, начиная с версии 4, существует понятие ссылок — жестких и символических. Имена всех переменных в PHP должны начинаться со знака `$` — так интерпретатору значительно легче "понять" и отличить их, например, в строках. Имена переменных чувствительны к регистру букв: например, `$var` — не то же самое, что `$Var` или `$VAR`: В официальной документации PHP указано, что имя переменной может состоять не только из букв "Латиницы" и цифр, но также и из любых символов, код ASCII которых старше 127, — в частности, и из символов кириллицы, то есть "русских" букв! Однако не рекомендуется применять кириллицу в именах переменных — хотя бы из-за того, что в различных кодировках ее буквы имеют различные коды. Впрочем, можно поэкспериментировать и делать так, как вам будет удобно. Можно сказать, что переменные в PHP — это особые объекты, которые могут содержать в буквальном смысле все, что угодно.



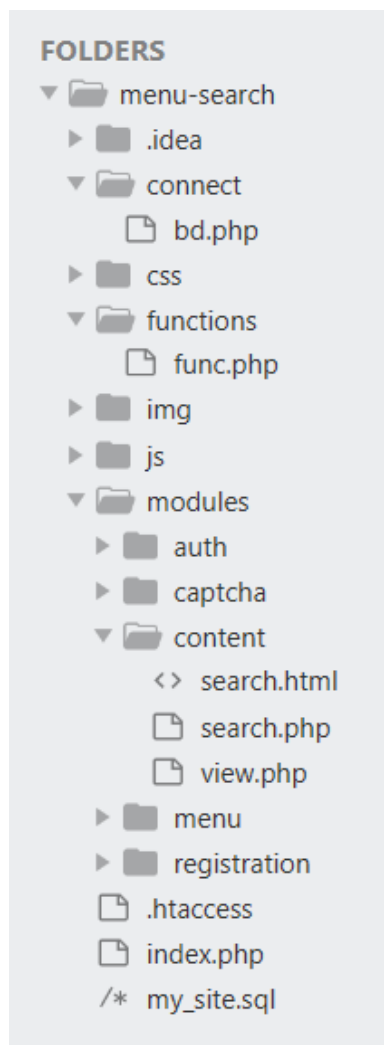
Тема лабораторной работы: Работа с символьными переменными

Материалы лабораторной работы: Если строка определяется в двойных кавычках, либо при помощи `heredoc`, переменные внутри нее обрабатываются. Существует синтаксис, дающий возможность обработки переменной, значения массива (`array`) или свойства объекта (`object`). Для этого используются `{}`. Если интерпретатор встречает знак доллара (`$`), он захватывает так много символов, сколько возможно, чтобы сформировать правильное имя переменной. Если вы хотите точно определить конец имени, заключайте имя переменной в фигурные скобки.

1.4. Интеграция вывода контента из БД в проект УМК

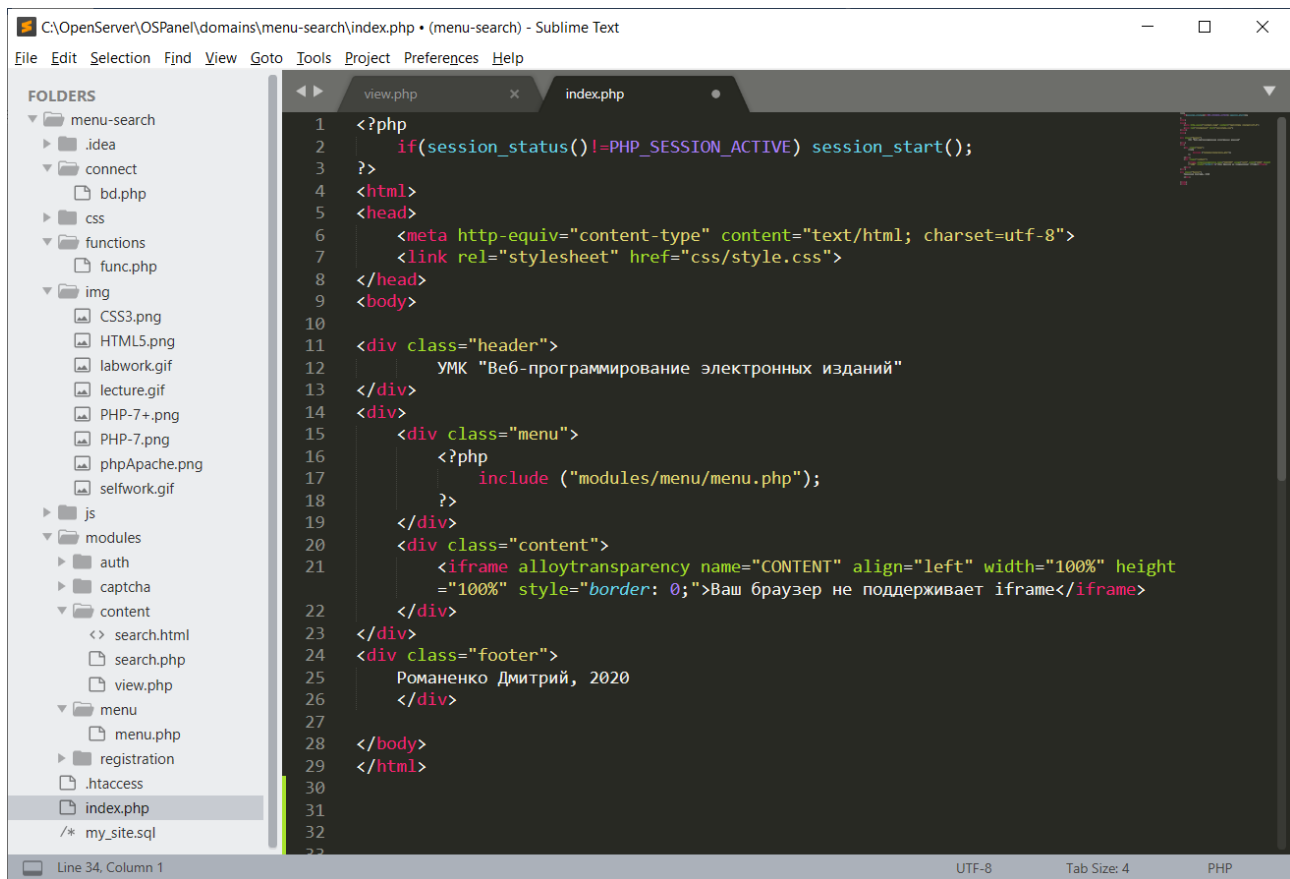
Рассмотрим процесс интеграции в проект УМК способа работы с контентом, представленного в пункте 1.2 (с использованием `iframe`).

И так, постепенно с реализацией функциональных возможностей начнет меняться структура проекта, а точнее приближаться к той, что была показана в первой лекции. На данном этапе она примет следующий вид.



Как видно, все основные функциональные модули будут располагаться каждый в своей папке. Модули авторизации (auth), регистрации (registration), формирования капчи (captcha) взяты из итогового проекта прошлого семестра, но только несколько адаптированы под данный проект. Модуль формирования меню расположен в каталоге menu. Используемые в проекте функции находятся в файле func.php, который хранится в отдельном каталоге functions, файл подключения к БД db.php хранится в каталоге connect. CSS-файлы хранятся в папке css, а вот javascript-файлы в каталоге js.

Полная структура проекта на данном этапе работы над ним представлена на следующем рисунке.



Далее приведем листинги некоторых из перечисленных файлов, важных для понимания дальнейшего процесса вывода контента.

Файл *menu.php* (меню проекта хранится в таблице *menu*, как было рассмотрено в соответствующей лекции).

```

<html>
    <head>
        <meta charset="windows-1251">
        <link rel="stylesheet" href="./css/style.css">
    </head>
    <style>
        ul {
            list-style: none; /*убираем маркеры списка*/
            margin: 5;
            margin-right: 20;
            padding-left: 10;

        }
        a {
            text-decoration: none; /*убираем подчеркивание текста
ССЫЛОК*/
        }
        li {
            margin-bottom: 10;
            margin-top: 10;

```



```

    }
</style>
</html>
<?php
    include './functions/func.php';
    //Выводим каталог на экран с помощью рекурсивной функции
    echo '<div style="padding:10px; border-bottom:1px solid #074776">';
    echo 'МЕНЮ САЙТА';
    echo '</div>';

    $result=get_category();

    echo "<div>";
    view_category($result);
    echo "</div>";
?>

```

Функция `get_category()`

```

function get_category()
{
    include './connect/bd.php';
    //запрос к базе данных
    $query ="SELECT * FROM menu";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    if($result)
    {
        $arr_cat = array();
        if(mysqli_num_rows($result) != 0)
        {
            //В цикле формируем массив
            for($i = 0; $i < mysqli_num_rows($result);$i++)
            {
                $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
                //Формируем массив, где ключами являются адишники на
родительские категории
                if(empty($arr_cat[$row['parent_id']])) {
                    $arr_cat[$row['parent_id']] = array();
                }
                $arr_cat[$row['parent_id']][] = $row;
            }
        }
        //возвращаем массив
        return $arr_cat;
    }
}

```

Функция `view_category()`

```
//вывод каталога с помощью рекурсии
function view_category($arr,$parent_id = 0)
{
    //Условия выхода из рекурсии
    if(empty($arr[$parent_id])) {
        return;
    }
    echo '<ul>';
    //перебираем в цикле массив и выводим на экран
    for($i = 0; $i < count($arr[$parent_id]);$i++) {
        echo '<li><a
href="./modules/content/view.php?category_id='.$arr[$parent_id][$i]['id']
.
        '&parent_id='.$parent_id.'"target="CONTENT">'
        . $arr[$parent_id][$i]['title'].'</a>';
        //рекурсия - проверяем нет ли дочерних категорий
        view_category($arr,$arr[$parent_id][$i]['id']);
        echo '</li>';
    }
    echo '</ul>';
}
```

Далее разработаем отдельный файл (скрипт) *view.php*, позволяющий выводить материалы лекций и лабораторных работ из соответствующих таблиц. Но прежде, чем приступить к его рассмотрению, необходимо обратить внимание на строку кода (выделена желтым), формирующую гиперссылку при отображении пунктов меню (т.е. функцию `view_category`).

```
function view_category($arr,$parent_id = 0)
{
    //Условия выхода из рекурсии
    if(empty($arr[$parent_id])) {
        return;
    }
    echo '<ul>';
    //перебираем в цикле массив и выводим на экран
    for($i = 0; $i < count($arr[$parent_id]);$i++) {
        echo '<li><a
href="./modules/content/view.php?category_id='.$arr[$parent_id][$i]['id']
.
        '&parent_id='.$parent_id.'"target="CONTENT">'
        . $arr[$parent_id][$i]['title'].'</a>';
        view_category($arr,$arr[$parent_id][$i]['id']);
    }
}
```

```
        echo '</li>';
    }
    echo '</ul>';
}
```

Как видно, теперь в код, формирующий гиперссылку, добавлен файл-обработчик view.php, а при его вызове методом GET будут передаваться два параметра: category_id и parent_id.

```
category_id='.$arr[$parent_id][$i]['id'].'&parent_id='.$parent_id.
```

Именно они позволят определить, какой пункт меню выбрал пользователь. Так, например, если parent_id=1, то пользователем выбраны лекции. А вот какую лекцию необходимо загружать, можно определить исходя из значения category_id.

Поэтому файл view.php по сути начинается с считывания переданных данных методом GET, их обработке, записи требуемых данных в сессию (наименования родительской категории выбранного пользователем пункта меню), записи в переменную \$wheresearch названия таблиц labworks или lectures, из которой необходимо выгружать информацию.

```
$category_id=$_GET['category_id'];
$parent_id=$_GET['parent_id'];

include '../connect/bd.php';
$query = "SELECT title FROM menu where id=$parent_id";
$result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    if ($result){
        $row=mysqli_fetch_row($result);
        $_SESSION['menu']=$row[0];
    }

    $query = "SELECT title FROM menu where id=$category_id";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    if ($result){
        $row=mysqli_fetch_row($result);
    }

    if ($_SESSION['menu']=='Лекции') $wheresearch='lectures';
    if ($_SESSION['menu']=='Лабораторные работы')
$wheresearch='labworks';
```

Далее выполняется выборка информации из соответствующей таблицы в соответствии с установленными (хранятся в сессии) параметрами.

```
$query = "SELECT name, content FROM $wheresearch WHERE id=$category_id";
```

```

        $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
        if($result)
        {
            $rows = mysqli_num_rows($result); // количество полученных
строк
            for ($i = 0 ; $i < $rows ; ++$i)
            {
                $row = mysqli_fetch_assoc($result);
                echo "<p>";
                echo "<b>";
                if ($_SESSION['menu']=='Лекции') {
                    echo "Тема лекции \"";
                } else {
                    echo "Тема лабораторной работы \"";
                }
                echo $row['name'];
                echo "\"";
                echo "</b>";
                echo "</p>";
                echo "<br>";
                echo $row['content'];
            }
        }

        // закрываем подключение
mysqli_close($link);

```

Вверху блока с выводимыми материалами разместим ссылку для организации поиска.

```

<div class='search'>
    <a href='search.html'>ПОИСК</a>
</div>

```

Также надо обработать ситуацию, когда пользователь выбрал родительскую категорию, например, *Лекции* или *Лабораторные работы* или наконец *Самостоятельная работа*. Признаком этого будет значение переменной `parent_id = 0`. Таким образом целесообразно изначально проверить значение данной переменной: если она не равно 0, то должны выполняться строки кода, отвечающие за вывод материалов лекций и лабораторных работ, а в противном случае, в зависимости от выбранной пользователем родительской категории будет выводиться соответствующая информация (например, в виде картинок).

В итоге весь файл `view.php` будет выглядеть следующим образом.

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="windows-1251">

```

```

        <link rel="stylesheet" href="./css/style.css">
</head>
<style>
    .search {
        margin-bottom: 20px;
        padding-bottom: 20px;
        border-bottom: solid;
        border-width: 1px;
        border-color: #66CCCC;
        text-align: right;
        padding-right: 80px;
    }
    a{
        text-decoration: none
    }
</style>
<div class='search'>
    <a href='search.html'>ПОИСК</a>
</div>

<div class='content'>
<?php
    $category_id=$_GET['category_id'];
    $parent_id=$_GET['parent_id'];

    include '../connect/bd.php';
    $query ="SELECT title FROM menu where id=$parent_id";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    if ($result){
        $row=mysqli_fetch_row($result);
        $_SESSION['menu']=$row[0];
    }

    $query ="SELECT title FROM menu where id=$category_id";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    if ($result){
        $row=mysqli_fetch_row($result);
    }

if ($parent_id!=0){
    if ($_SESSION['menu']=='Лекции') $wheresearch='lectures';
    if ($_SESSION['menu']=='Лабораторные работы')
$wheresearch='labworks';

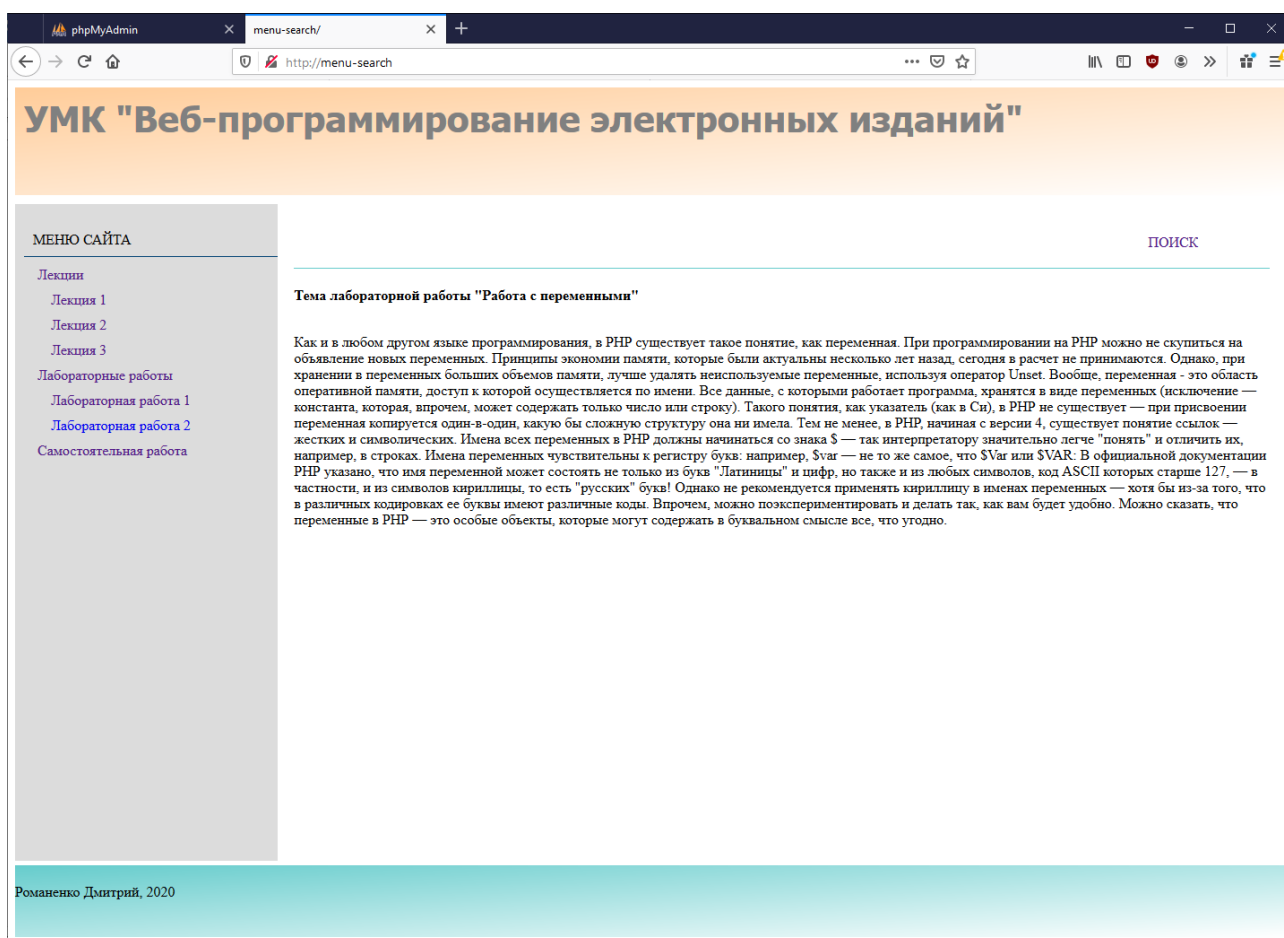
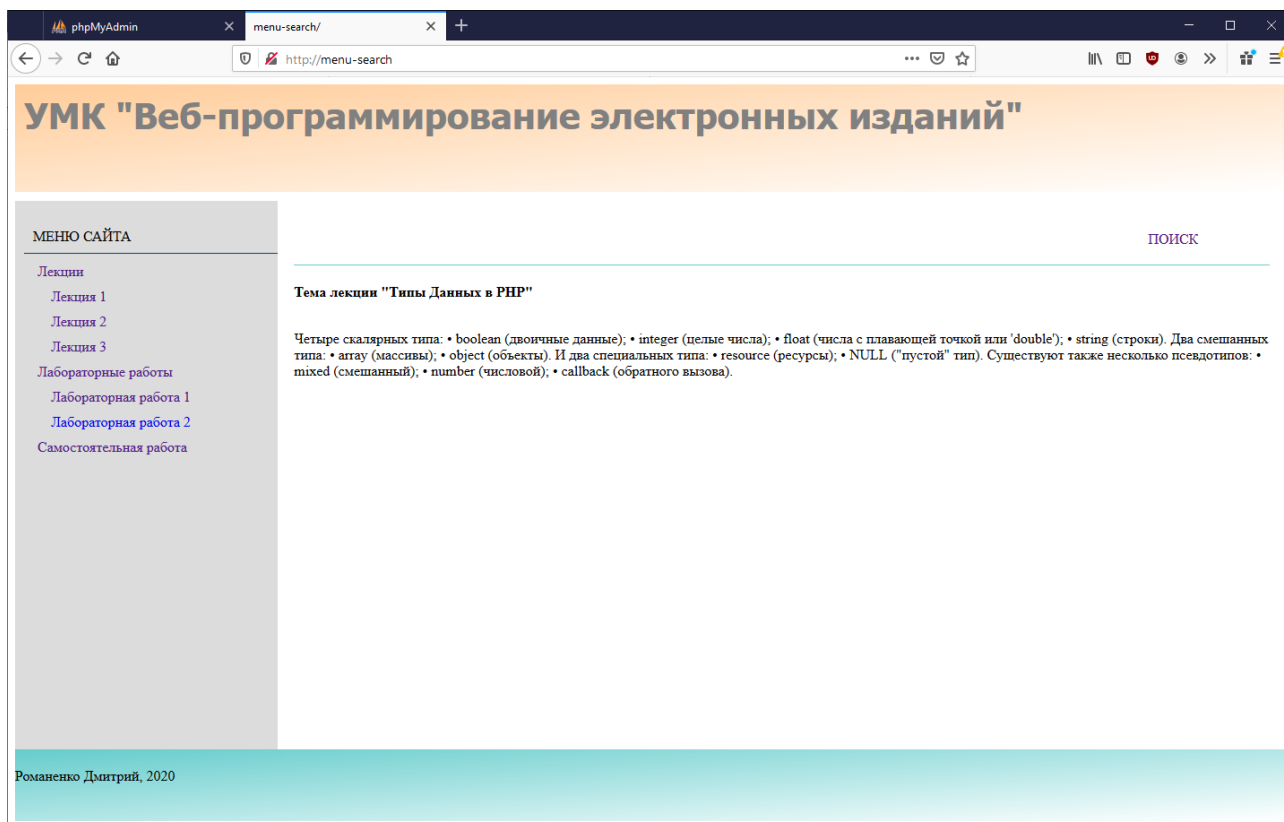
```

```

        $query = "SELECT name, content FROM $wheresearch WHERE
id=$category_id";
        $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
        if($result)
        {
            $rows = mysqli_num_rows($result); // количество полученных
строк
            for ($i = 0 ; $i < $rows ; ++$i)
            {
                $row = mysqli_fetch_assoc($result);
                echo "<p>";
                echo "<b>";
                if ($_SESSION['menu']=='Лекции') {
                    echo "Тема лекции \"";
                } else {
                    echo "Тема лабораторной работы \"";
                }
                echo $row['name'];
                echo "\"";
                echo "</b>";
                echo "</p>";
                echo "<br>";
                echo $row['content'];
            }
        }
        // закрываем подключение
        mysqli_close($link);
    }
else
{
    $query = "SELECT link FROM menu where id=$category_id";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    if ($result)
    {
        $row=mysqli_fetch_row($result);
        echo "<img src='../..'/img/".$row[0]."'>";
    }
    mysqli_close($link);
}
?>
</div>
</html>

```

А сама страница с меню и выведенным контентом, будет выглядеть следующим образом.



Если пользователь выбрал родительскую категорию «Лекции», то вместо материалов выводится просто картинка, хранящаяся в папке *img*.

phpMyAdmin

menu-search/

+

← → ↻ 🏠

🔒 http://menu-search

⋮ 🛡️ ☆

🔍 📄 📱 📧 📧 📧 📧 📧 📧

УМК "Веб-программирование электронных изданий"

МЕНЮ САЙТА

Лекции

Лекция 1

Лекция 2

Лекция 3

Лабораторные работы


Лабораторная работа 1

Лабораторная работа 2

Самостоятельная работа

ПОИСК

LECTURES



Романенко Дмитрий, 2020

2. Организация поиска средствами PHP и MySQL

2.1 Организация поиска с перезагрузкой страницы.

Рассмотрим краткий алгоритм реализации поиска (схематичное представление алгоритма приведено на рис.1).

Форма поиска открывается при щелчке пользователя по ссылке ПОИСК. Введя необходимые параметры поиска, пользователь щелкает по соответствующей кнопке, тем самым выполняя POST запрос из формы поиска, этот запрос передается специальному скрипту-обработчику, который должен обработать поисковый запрос пользователя и вернуть результат.

Сначала скрипт должен обработать должным образом запрос пользователя для обеспечения безопасности, затем выполняется запрос к базе данных (код подключения к БД, как обычно, вынесен в отдельный файл), который возвращает в ассоциативном массиве результаты, которые должны будут выводиться на экран.

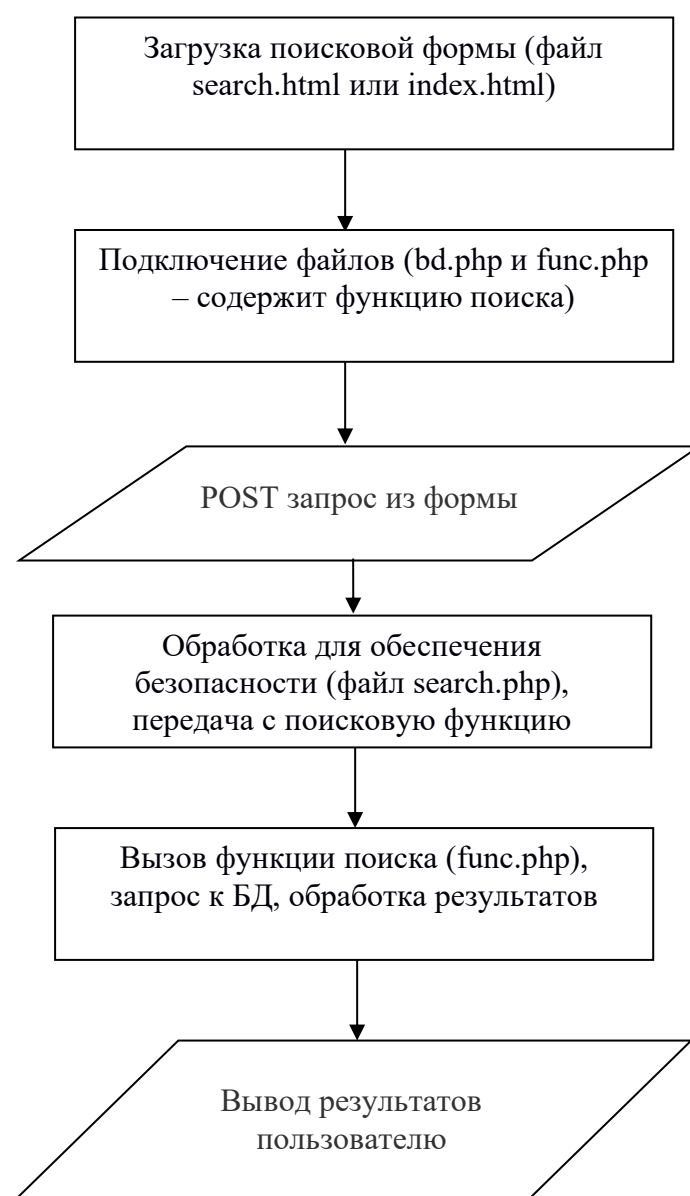
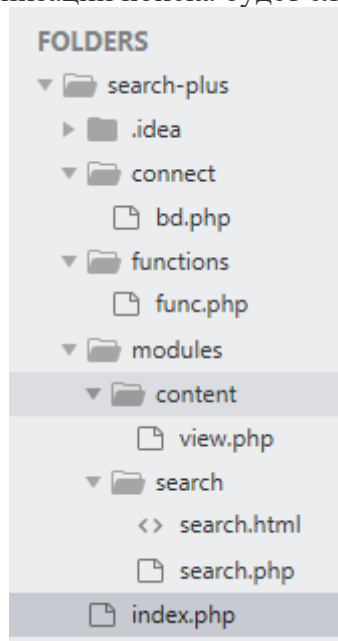


Рис.1. Алгоритм организации поиска информации

Общая структура проекта организации поиска будет следующая.



Отметим, что данная структура аналогична ранее рассматриваемым проектам. Далее рассмотрим более подробно некоторые файлы.

Для начала создадим форму поиска в виде отдельного файла (search.html), который будет подключаться на главной странице проекта (в файле index.php):

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <meta http-equiv="content-type" content="text/html; charset=utf-
8">
</head>
<style>
  div {
    margin-top: 30px;
    text-align: center;
  }
</style>
<body>
<div>
  <form name="search" method="post"
action="../../modules/search/search.php">
    <input type="search" name="query" placeholder="Поиск"
size="29">
    <p>-----</p>
    <p><b>&emsp;&emsp;Параметры поиска</b></p>
    <p><input name="choice" type="radio" value="lec" checked> По
лекционным материалам</p>
    <p><input name="choice" type="radio" value="lab" > По
лабораторным материалам</p>
    <p>-----</p>
```

```
<button type="submit">Найти</button>
</form>
</div>
</body>
</html>
```

Для пользователя это будет выглядеть следующим образом.

Параметры поиска
☒ По лекционным материалам
☐ По лабораторным материалам

Эта форма и будет отправлять сам поисковый запрос скрипту **search.php** (находится в той же папке, что и поисковая форма). Данный скрипт-обработчик будет выглядеть следующим образом.

```
<?php
    include '../connect/bd.php';
    include '../functions/func.php';

    if (!empty($_POST['query']))
    {
        if ($_POST['choice']=="lec") $table='lectures'; Else
        $table='labworks';
        $search_result = search ($link, $_POST['query'],$table);
        echo $search_result;
    }
?>
```

Данный из формы передаются в скрипт-обработчик методом POST.

Непосредственно поиск по БД будет осуществляться отдельной разработанной функцией **search** (так будет удобнее в дальнейшем интегрироваться в общую систему УМК).

```
function search ($link, $what, $table)
{
}
}
```

Данная функция должна изначально «обработать» получаемую из поискового поля строку, чтобы запрос стал «безопасным». Такую обработку нужно делать обязательно, т. к. любая форма на сайте — это потенциальная уязвимость для злоумышленников.

```
$what = trim($what);  
$what = mysqli_real_escape_string($link, $what);  
$what = htmlspecialchars($what);
```

Приведем некоторые пояснения к используемым функциям.

`htmlspecialchars()`

В HTML некоторые символы имеют специальное значение и для сохранения своего значения должны быть преобразованы в HTML сущности. Эта функция возвращает строку, над которой проведены некоторые из таких преобразований. Эта функция полезна при отображении данных, введенных пользователем, которые могут содержать нежелательные HTML тэги.

Примеры преобразования:

- `'&'` (амперсанд) преобразуется в `'&'`
- `'\"'` (двойная кавычка) преобразуется в `'"'` when `ENT_NOQUOTES` is not set.
- `'\"'` (одиночная кавычка) преобразуется в `'''` только в режиме `ENT_QUOTES`.
- `'<'` (знак "меньше чем") преобразуется в `'<'`
- `'>'` (знак "больше чем") преобразуется в `'>'`

`mysqli_real_escape_string($link, $query)`

Экранирует специальные символы в строке для использования в SQL выражении, используя текущий набор символов соединения. Эта функция используется для создания допустимых в SQL строк, которые можно использовать в SQL выражениях. Заданная строка кодируется в экранированную SQL строку, используя текущий набор символов подключения. Набор символов должен быть задан либо на стороне сервера, либо с помощью API функции `mysqli_set_charset()`.

Необходимо обратить внимание, что форма записи данной функции для процедурного стиля программирования требует использование двух переменных – содержащих строку подключения и строку с текстом запроса.

```
trim ( string $str [, string $character_mask = " \t\n\r\0\x0B" ] )
```

Эта функция возвращает строку `str` с удаленными из начала и конца строки пробелами. Если второй параметр не передан, `trim()` удаляет следующие символы:

- `" "` (ASCII 32 (0x20)), обычный пробел.
- `"\t"` (ASCII 9 (0x09)), символ табуляции.
- `"\n"` (ASCII 10 (0x0A)), символ перевода строки.
- `"\r"` (ASCII 13 (0x0D)), символ возврата каретки.
- `"\0"` (ASCII 0 (0x00)), NUL-байт.
- `"\x0B"` (ASCII 11 (0x0B)), вертикальная табуляция.

Далее введем проверку на минимальную и максимальную длину поисковой комбинации слов (переменная `$text` в дальнейшем будет использована для возврата необходимой информации для пользователя из функции в основную часть программы).

```
if (strlen($what) < 3)  
{  
    $text = '<p>Слишком короткий поисковый запрос.</p>';
```

```

    }
    else if (strlen($what) > 128)
    {
        $text = '<p>Слишком длинный поисковый запрос.</p>';
    }
    else{

    }
}

```

Далее формируется SQL запрос, который и будем выполнять поиск соответствующей информации в БД, и наконец он выполняется. Так, например, при поиске по таблице с лекциями он будет выглядеть следующим образом.

```

$query = "SELECT `id`, `name`, `content` FROM $table WHERE `content` LIKE
'%" . $what . "%' OR `name` LIKE '%" . $what . "%'";
$result = mysqli_query($link, $query) or die("О ш и б к а " .
mysqli_error($link));

```

Далее, по сути, должны быть написаны строки обработки результатов запроса, которые будут решать следующие задачи:

1. Подсчет количество полученных записей (строк).
2. В виду того, что в БД может храниться текстовая информация любой длины (количество символов), то предполагается вывод на экран лишь ее небольшой части и гиперссылки на полный вариант найденной информации.
3. Так как для организации поиска будет использована функция, то вся информация, подлежащая выводу на экран, последовательно упаковывается в одну переменную текстового типа (сразу со всеми необходимыми тегами), которая и будет отправлена после выполнения возврата из функции через *echo* на экран пользователя.

```

if($result){
    $rows = mysqli_num_rows($result); // количество полученных строк
    $text = '<p>Всего по запросу <b>'. $what . '</b> найдено совпадений:
' . $rows . '</p>';
    $text.='-----';
    $text.='-----';
    $text.= "<br>";

    for ($i = 0 ; $i < $rows ; ++$i){
        $row = mysqli_fetch_assoc($result);
        $id=$row['id'];
        $text.= "<p>";
        $text.= "<strong>";
        if ($table=="lectures") $text.= "Тема лекции \"; else $text.=
"Тема лабораторного занятия \";

        $text.= $row['name'];
        $text.= "\"";
        $text.= "</strong>";
        $text.= "</p>";
    }
}

```

```

        $text.= substr($row[content],0,240).' ...';
        if ($table=="lectures"){
            $text.= "<a
href='../..'/modules/content/view.php?where=$table&what=$id'>Смотреть
полный текст</a>";
        }
        else{
            $text.= "<a
href='../..'/modules/content/view.php?where=$table&what=$id'>Смотреть
полный текст</a>";
        }
        $text.= "<br>";
        $text.= "<br>";
    }
}

```

Таким образом полный код файл функции поиска будет следующим. Необходимо обратить внимание, что в функцию будут передавать три переменные: переменная, хранящая подключение (**\$link**), переменная (**\$what**), хранящая строку что пользователь ищет, а также переменная (**\$table**), хранящая наименование таблицы, в которой будет осуществляться поиск).

```

<?php

function search ($link, $what, $table)
{
    $what = trim($what);
    $what = mysqli_real_escape_string($link, $what);
    $what = htmlspecialchars($what);

    if (!empty($what))
    {
        if (strlen($what) < 3)
        {
            $text = '<p>Слишком короткий поисковый запрос.</p>';
        }
        else if (strlen($query) > 128)
        {
            $text = '<p>Слишком длинный поисковый запрос.</p>';
        }
        else
        {
            $query = "SELECT `id`,`name`,`content` FROM $table WHERE
`content` LIKE '%$what%' OR `name` LIKE '%$what%'";
            $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));

```

```

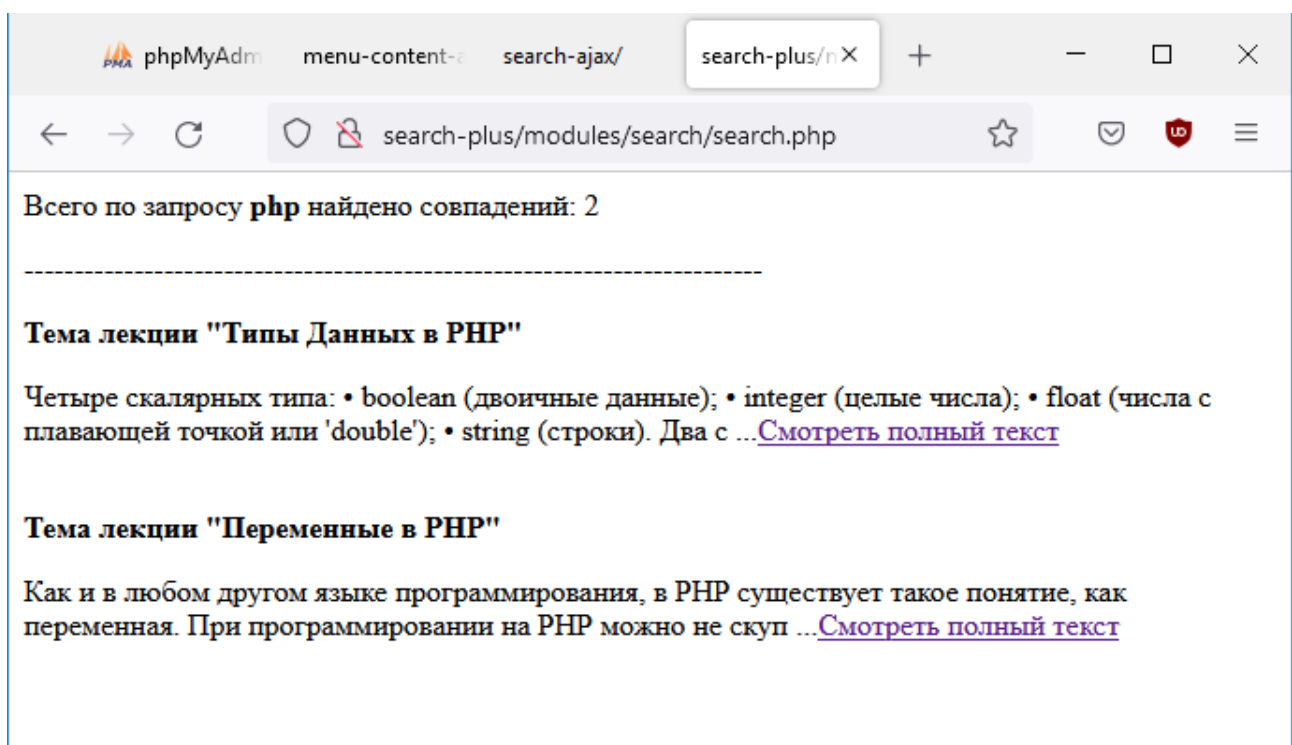
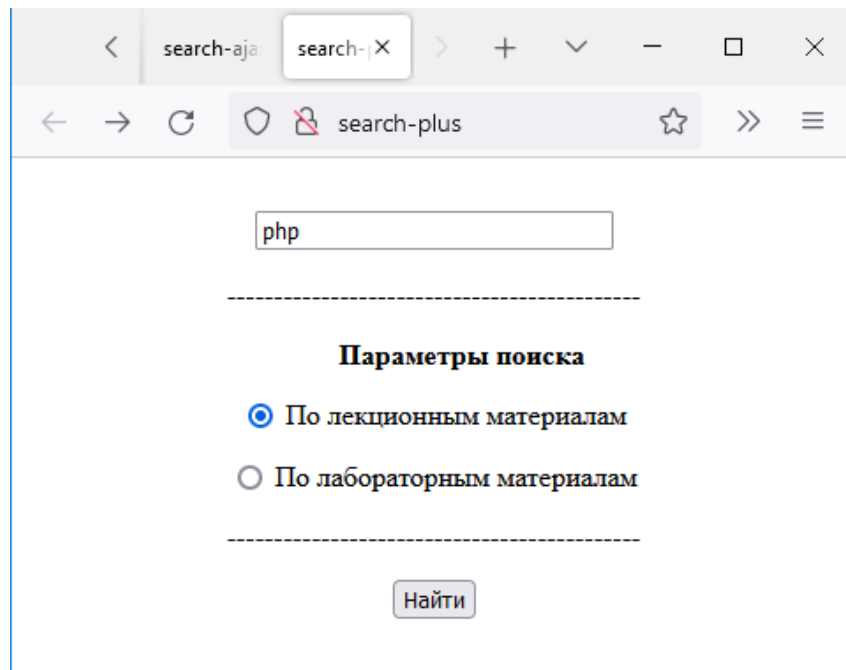
        if($result)
        {
            $rows = mysqli_num_rows($result); // количество
полученных строк
            $text = '<p>Всего по запросу <b>'.$what.'</b> найдено
совпадений: '.$rows.'</p>';
            $text.='-----
-----';
            $text.= "<br>";

            for ($i = 0 ; $i < $rows ; ++$i)
            {
                $row = mysqli_fetch_assoc($result);
                $id=$row['id'];
                $text.= "<p>";
                $text.= "<b>";
                if ($table=="lectures") $text.= "Тема лекции \";
else $text.= "Тема лабораторного занятия \";

                $text.= $row['name'];
                $text.= "\"";
                $text.= "</b>";
                $text.= "</p>";
                $text.= substr($row[content],0,240).' ...';
                if ($table=="lectures"){
                    $text.= "<a
href='../..'/modules/content/view.php?where=$table&what=$id'>Смотреть
полный текст</a>";
                }
                else{
                    $text.= "<a
href='../..'/modules/content/view.php?where=$table&what=$id'>Смотреть
полный текст</a>";
                }
                $text.= "<br>";
                $text.= "<br>";
            }
        }
    }
    return $text;
}
//конец функции поиска

?>

```



Как видно из последнего скриншота, сразу пользователю выводится лишь фрагмент материала, в котором было найдена введенная пользователем последовательность, а также сформирована гиперссылка, которая вызывает файл, предназначенный для отображения контента (view.php). По данной ссылке методом GET передается два параметра: наименование таблицы БД, в которой данный материал хранится (переменная \$table) и id записи в данной таблице (переменная id).


```
"<a href='.././modules/content/view.php?where=$table&what=$id'>Смотреть  
полный текст</a>
```

Остается извлечь из требуемой таблицы требуемую запись и отобразить ее пользователю
– данную задачу решает файл view.php.

```
<?php  
    if(session_status() != PHP_SESSION_ACTIVE) session_start();  
?>  
<?php  
    include '.././connect/bd.php';  
  
    $table=$_GET['where'];  
    $what=$_GET['what'];  
  
    if ($table=='lectures'){  
        $query = "SELECT * FROM $table WHERE id='$what'";  
        $result = mysqli_query($link, $query) or die("Ошибка " .  
mysqli_error($link));  
        if($result)  
        {  
            $row = mysqli_fetch_assoc($result);  
  
            echo "<p>";  
            echo "<b>";  
            echo "Тема лекции \";  
            echo "</b>";  
            echo $row['topic'];  
            echo "\"";  
            echo "</p>";  
  
            echo "<p>";  
            echo "<b>";  
            echo "Название лекции \";  
            echo "</b>";  
            echo $row['name'];  
            echo "\"";  
            echo "</p>";  
  
            echo "<p>";  
            echo $row['content'];  
            echo "</p>";  
        }  
        mysqli_close($link);  
    }  
}
```

```

        if ($table=='labworks'){
            $query = "SELECT * FROM $table WHERE id='$what'";
            $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
            if($result)
            {
                $row = mysqli_fetch_assoc($result);

                echo "<p>";
                echo "<b>";
                echo "Тема лабораторной работы \";
                echo "</b>";
                echo $row['name'];
                echo "\"";
                echo "</p>";

                echo "<p>";
                echo $row['content'];
                echo "</p>";
            }
            mysqli_close($link);
        }
?>

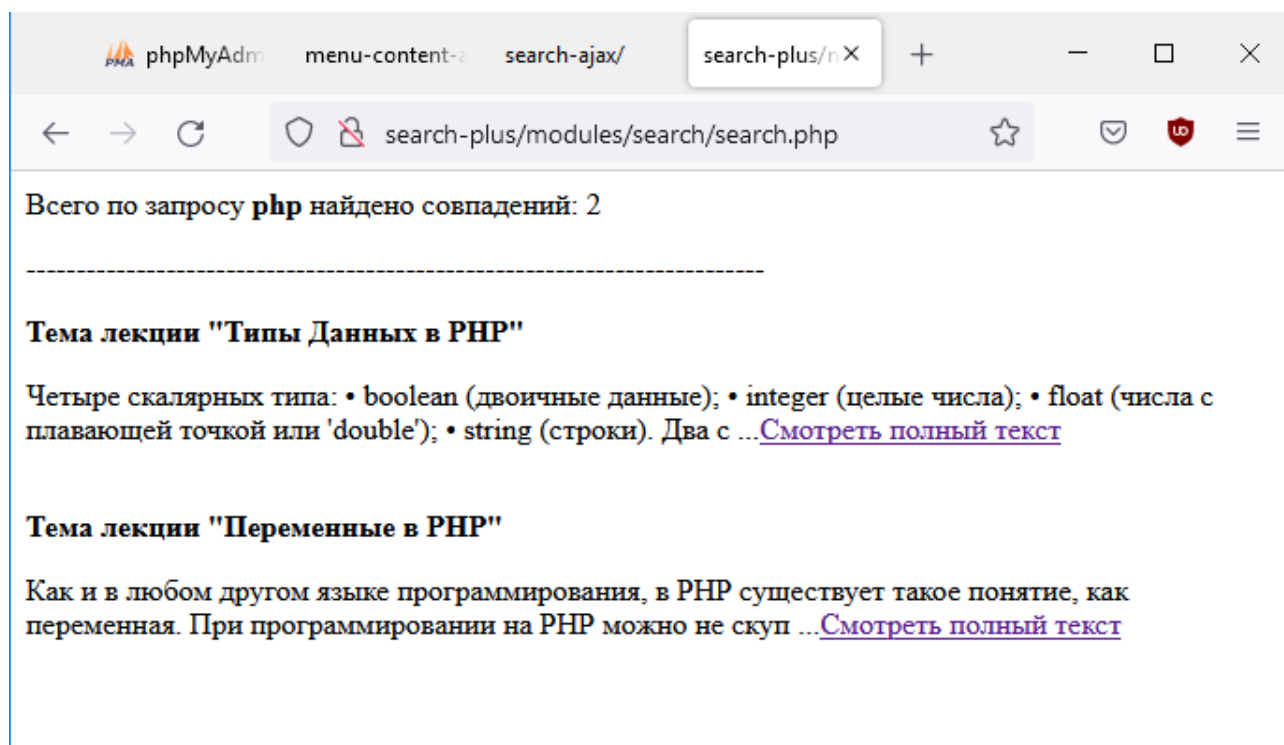
```

В итоге получим следующую последовательность отображения информации для пользователя:

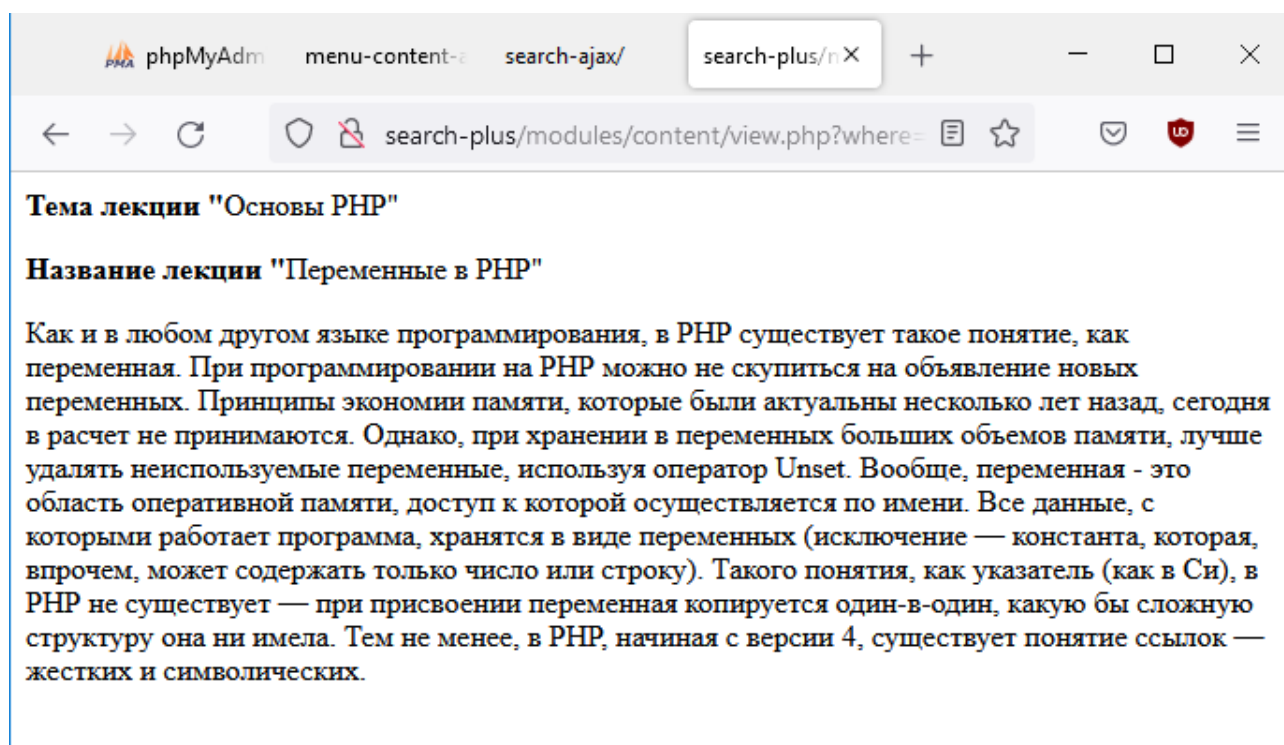
➤ Поисковая форма.

The screenshot shows a web browser window with the address bar displaying 'search-plus'. The page content includes a search input field with the text 'php'. Below the input field is a dashed line, followed by the heading 'Параметры поиска'. Under this heading are two radio buttons: the first is selected and labeled 'По лекционным материалам', and the second is unselected and labeled 'По лабораторным материалам'. Another dashed line follows, and at the bottom is a button labeled 'Найти'.

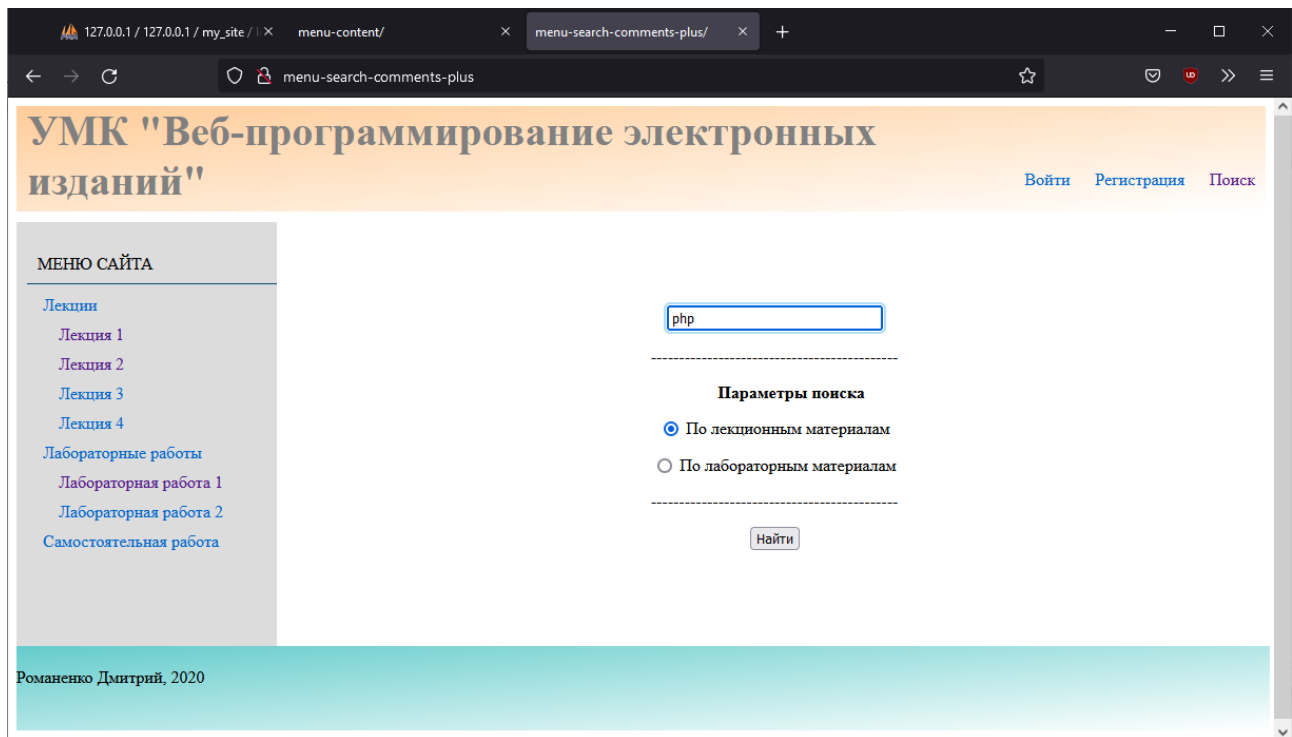
- Результаты поиска со ссылками на развернутое отображение материалов.



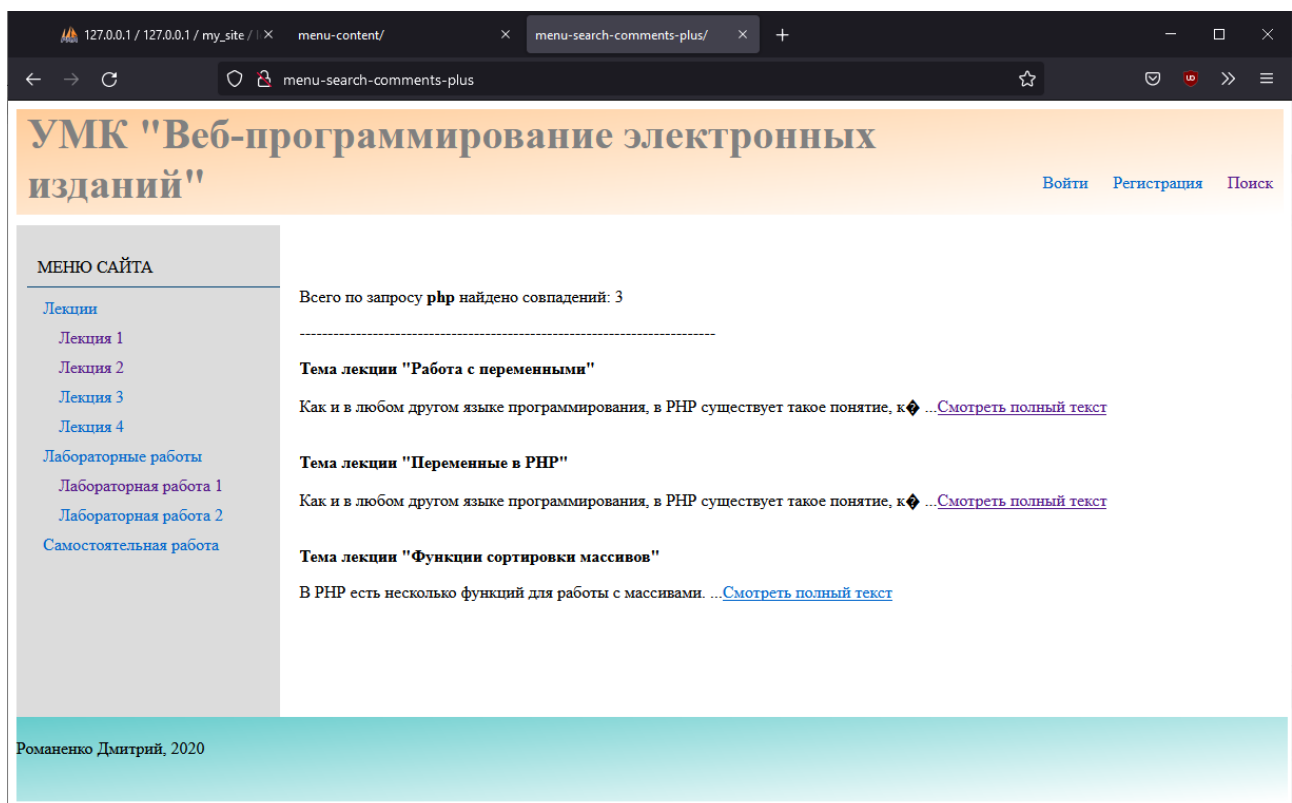
- Отображение материалов ресурса.



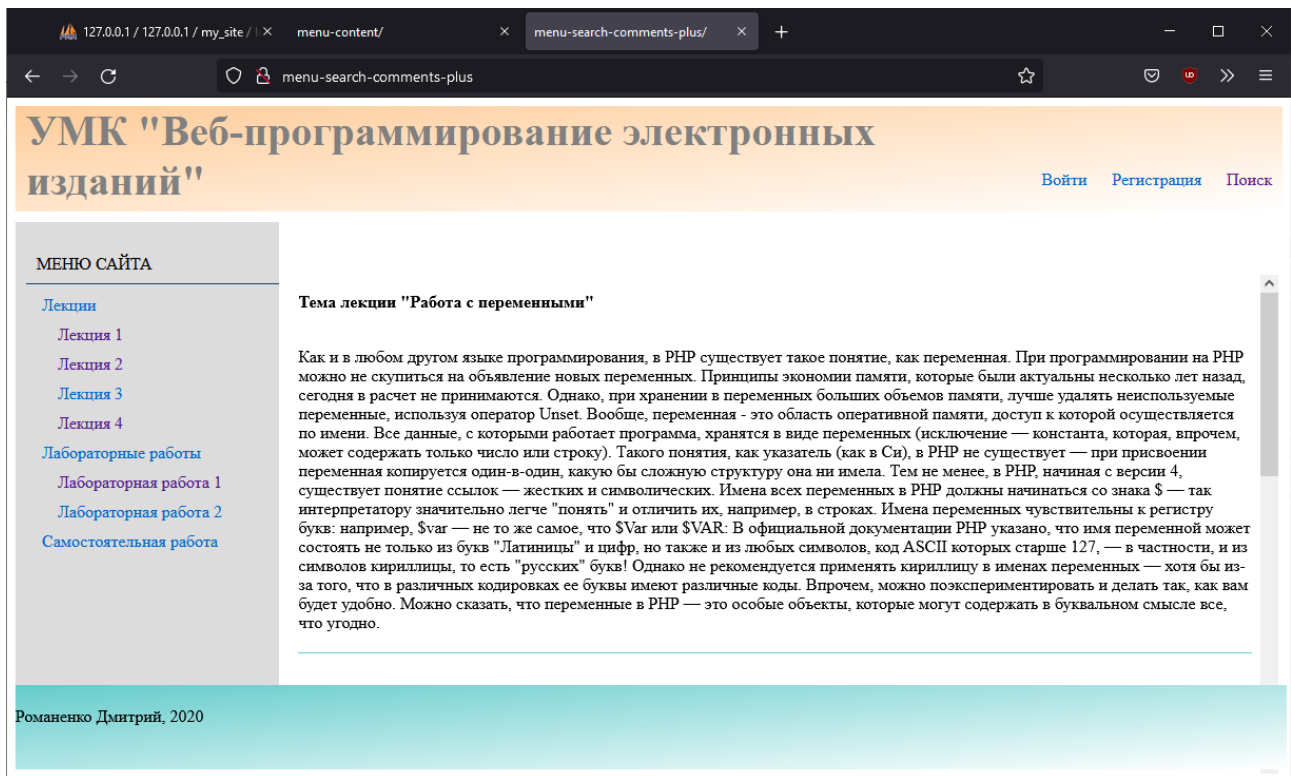
Процесс интеграции разработанного поиска в проект УМК фактически представляет собой перенос функции поиска в файл func.php, размещение файлов (каталогов) с поисковой формой и файлом обработчиком в модуле content (либо content и search), файл, отвечающий за отображение учебных материалов фактически уже имеется в проекта (использовался для обработки меню), ну и наконец в зависимости от выбранного варианты выхода на поисковую форму реализация соответствующих гиперссылок.



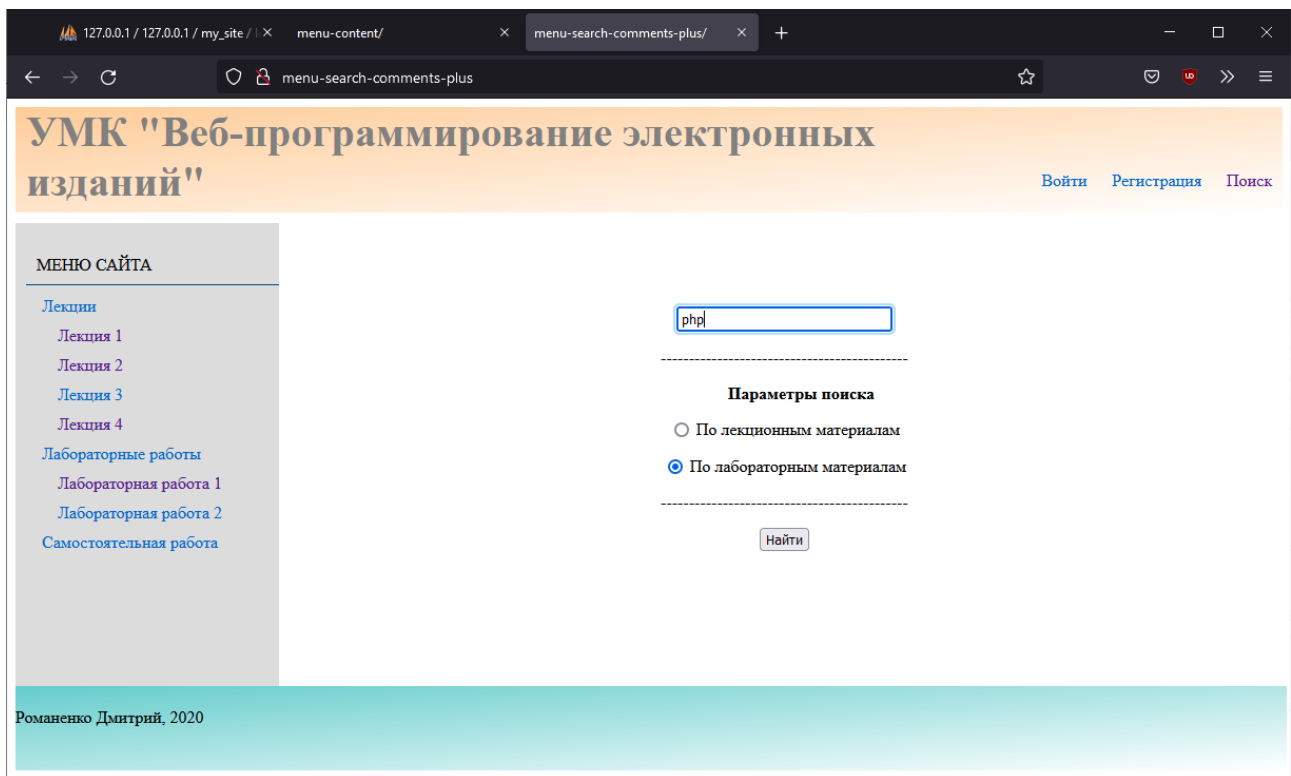
Результаты поиска отобразятся следующим образом.

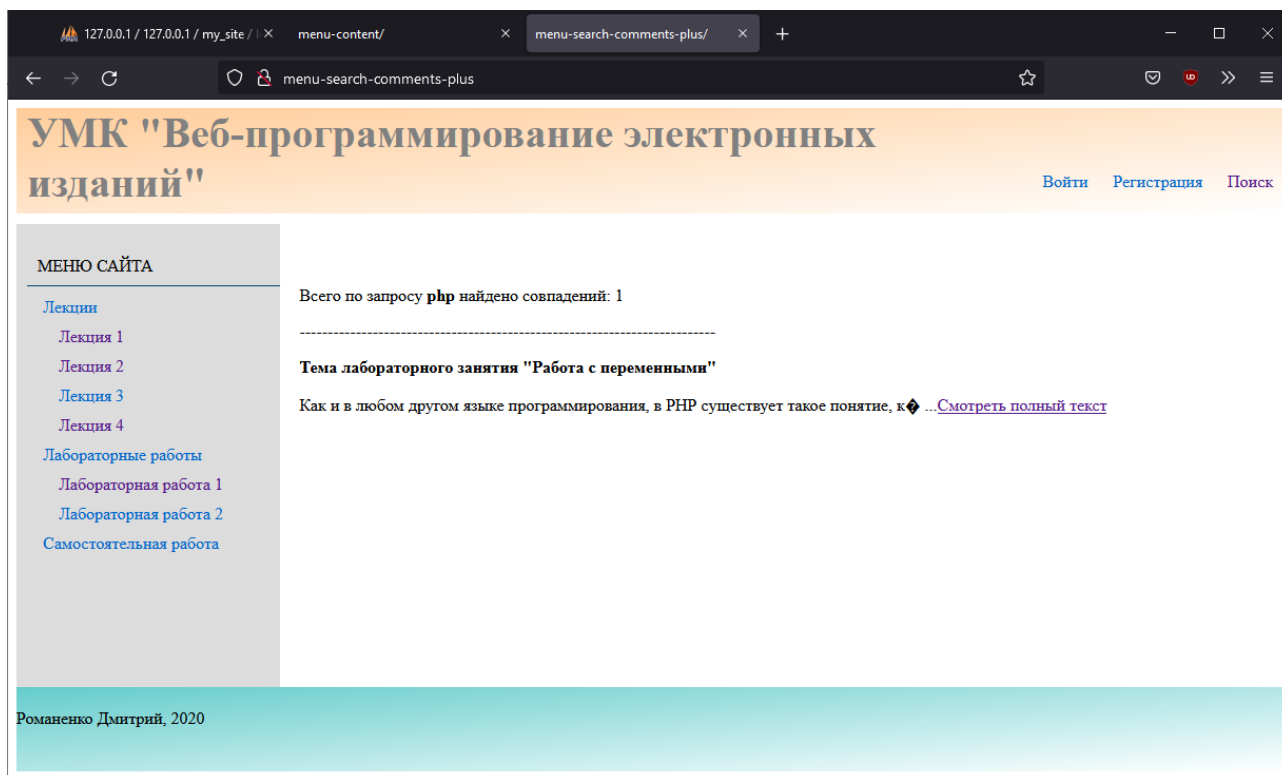


После щелчка по ссылке «смотреть полный текст» откроется соответствующий материал.



Аналогично все работает и по лабораторным материалам.



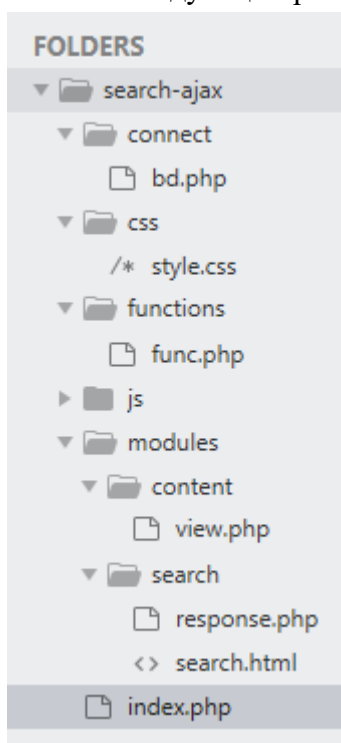


2.2. Организация поиска с использованием AJAX.

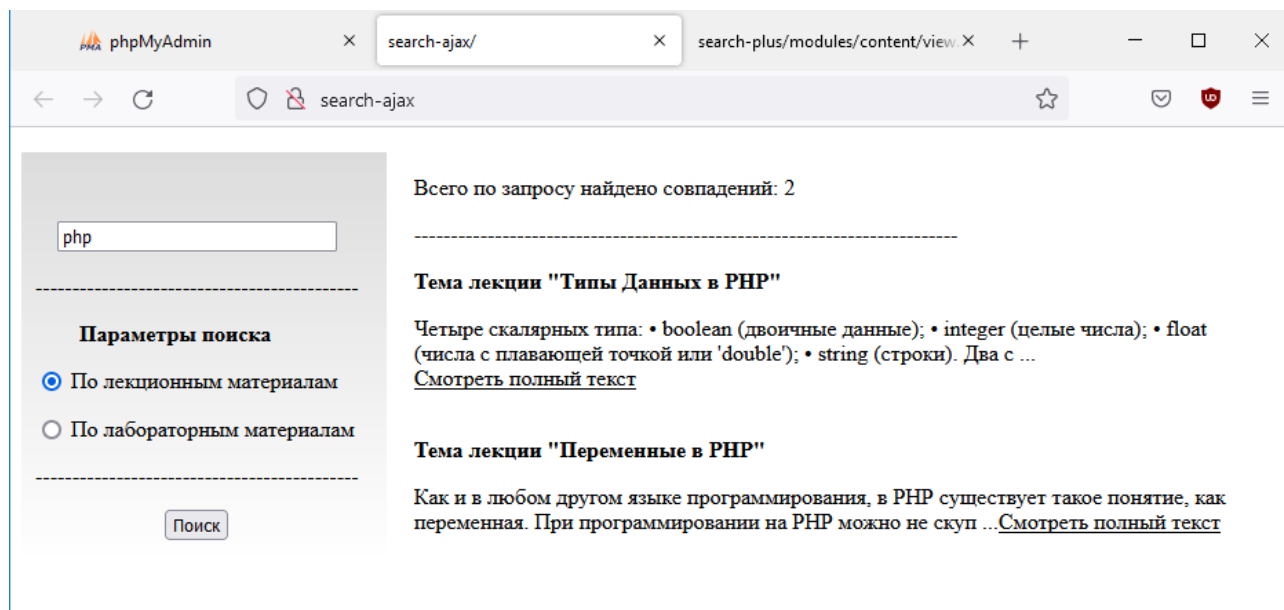
Рассмотрим вариант реализации поиска с использованием асинхронных принципов на основе технологии AJAX.

Отметим, что в целом алгоритм реализации поиска соответствует представленному в пункте 2.1 алгоритму, однако будет отличаться техника исполнения – будут задействованы библиотека jQuery и асинхронные процессы.

Структура проекта представлена на следующем рисунке.



Начнем со стартового файла – **index.php**. Его выполнение визуально будет выглядеть следующим образом.



Как видно из представленного скриншота, имеются два блока `div` – один предназначен для отображения поисковой формы, а второй – для отображения результатов поиска.

```
<div class="main">
    <?php
        include 'modules/search/search.html';
    ?>

</div>
<div class="content">

</div>
```

В блоке `main` реализовано подключение файла с поисковой формой, которая аналогична предыдущему примеру.

Также в данный файл необходимо добавить JS-функцию, предназначенную для считывания с формы параметров поиска и ajax-функцию, предназначенную для выполнения асинхронного запроса к БД для поиска данных (в ajax-функции вызывается файл `'modules/search/response.php'`)

```
<script language="javascript" type="text/javascript">
    $(function()
    {
        $(".search_button").click(function()
        {
            const form = document.getElementById("search");
            what=form.elements["what_search"].value;
```

```

        where=form.elements["where_search"].value;

        $.ajax({
            url:
'modules/search/response.php?where='+where+'&what='+what,
            success: function(data){
                $('.content').html(data);
            }
        });
    });
});
</script>

```

Полный листинг кода файла **index.php** в таком случае будет следующим.

```

<!DOCTYPE html>
<html lang="ru" >
    <head>
        <meta http-equiv="content-type" content="text/html; charset=utf-
8">

        <link rel="stylesheet" href="css/style.css">
        <script src="js/jquery.min.js" type="text/javascript"></script>
    </head>
    <style>
        ul {
            list-style: none;
            margin: 5;
            margin-right: 20;
            padding-left: 10;
            padding-right: 20px;
        }

        li {
            margin-bottom: 10;
            margin-top: 10;
        }
    </style>
    <body>
        <script language="javascript" type="text/javascript">
            $(function()
            {
                $(".search_button").click(function()
                {
                    const form = document.getElementById("search");
                    what=form.elements["what_search"].value;
                    where=form.elements["where_search"].value;

                    $.ajax({

```



```

        url:
'modules/search/response.php?where='+where+'&what='+what,
        success: function(data) {
            $('<div class="content">
            $.content').html(data);
        }
    });

});

});
</script>

<div class="main">
    <?php
        include 'modules/search/search.html';
    ?>
</div>
<div class="content">

</div>
</body>
</html>

```

Теперь рассмотрим код файла 'modules/search/response.php, который фактически отвечает за сам процесс поиска необходимой информации в БД. Сначала должны быть считаны необходимые для sql-запроса параметры, а затем вызвана функция поиска, описанная в предыдущем примере (в пункте 2.1).

```

<?php
//скрипт обработки поискового запроса
include '../connect/bd.php';
include '../functions/func.php';
$table=$_REQUEST['where'];
$what=$_REQUEST['what'];

echo search($link, $what, $table);
?>

```

Отметим, что код функции поиска будет следующим (изменения по сравнению с продемонстрированным кодом в пункте 2.1 заключаются в использовании для перехода на отображение полного варианта найденных материалов не гиперссылок, а кнопок (код выделен желтым цветом). Через атрибуты кнопок передаются дополнительные параметры – наименование таблицы в БД и id-записи в данной таблице, которую надо извлечь и отобразить.

```

<?php
// функция поиска
function search ($link, $what, $table)
{
    $what = trim($what);
    $what = mysqli_real_escape_string($link, $what);
    $what = htmlspecialchars($what);

    if (!empty($what))
    {
        if (strlen($what) < 3)
        {
            $text = '<p>Слишком короткий поисковый запрос.</p>';
        }
        else if (strlen($what) > 128)
        {
            $text = '<p>Слишком длинный поисковый запрос.</p>';
        }
        else
        {
            $q = "SELECT `id`, `name`, `content`
                FROM $table WHERE `content` LIKE '%$what%'
                OR `name` LIKE '%$what%'";
            $result = mysqli_query($link, $q) or die("Ошибка " .
mysqli_error($link));

            if($result)
            {
                $rows = mysqli_num_rows($result); // количество
полученных строк
                $text = '<p>Всего по запросу <b>'.$query.'</b>
найдено совпадений: '.$rows.'</p>';
                $text.='-----
-----';
                $text.= "<br>";

                for ($i = 0 ; $i < $rows ; ++$i)
                {
                    $row = mysqli_fetch_assoc($result);
                    $id=$row['id'];
                    $text.= "<p>";
                    $text.= "<strong>";
                    if ($table=="lectures") $text.= "Тема лекции
\""; else $text.= "Тема лабораторного занятия \";

                    $text.= $row['name'];
                    $text.= "\"";
                    $text.= "</strong>";

```

```

        $text.= "</p>";
        $text.= substr($row['content'],0,240).' ...';
        $text.= "<button name=\".$table.\"
value=\".$id.\" onClick='button_click(this) '>Смотреть полный
текст</button>";

        $text.= "<br>";
        $text.= "<br>";
    }
}
}
return $text;
}
//конец функции поиска
?>

```

Обработка щелчка по кнопке может быть реализована также с помощью JS.

```

function button_click(clicked_object)
{
    var table=clicked_object.getAttribute('name');
    var what=clicked_object.getAttribute('value');
    if (table=='lectures') content_lectures(what);
    if (table=='labworks') content_labworks(what);
}

```

Как видно из представленного кода, в зависимости от значения сформированной на основе атрибута кнопки переменной `table` далее реализуется вызов одной из функций – `content_lectures()` или `content_labworks()`, реализующих асинхронную выборку необходимых данных (материалов лекций или лабораторных работ) из таблицы.

Коды JS-функция `content_lectures()` и `content_labworks()` в таком случае следующими.

```

function content_lectures (id)
{
    $.ajax({
        dataType: 'json',
        url:
'../../modules/content/view.php?where=lectures&what='+id,
        success: function(jsondata)
        {
            $('<div>.content').html('<b>Тема лекции:</b> ' +
jsondata.topic + '<br><b>Название лекции:</b> ' + jsondata.name+
'<br><b>Материалы лекции:</b> ' + jsondata.content);
        }
    });
}

```

```

};

function content_labworks (id)
{
    $.ajax({
        dataType: 'json',
        url:
'../../modules/content/view.php?where=labworks&what='+id,
        success: function(jsondata)
        {
            $('<div>.content</div>').html('<b>Тема лабораторной работы:</b> '
+ jsondata.name + '<br><b>Материалы лабораторной работы:</b> ' +
jsondata.content);
        }
    });
};

```

Полный листинг файла modules/content/view.php будет следующим.

```

<!DOCTYPE html>
<html lang="ru" >
    <style>
        button {
            padding: 0;
            border: none;
            font: inherit;
            color: inherit;
            background-color: transparent;
            cursor: pointer;
            text-decoration: underline;
        }
    </style>

    <?php
//скрипт обработки поискового запроса
    include '../../connect/bd.php';
    include '../../functions/func.php';
    $table=$_REQUEST['where'];
    $what=$_REQUEST['what'];

    echo search($link, $what, $table);
?>

<script language="javascript" type="text/javascript">
/*JS-скрипт обработки щелчка для отображения полного текста учебных
материалов*/
    function button_click(clicked_object)
    {

```

```

        var table=clicked_object.getAttribute('name');
        var what=clicked_object.getAttribute('value');
        if (table=='lectures') content_lectures(what);
        if (table=='labworks') content_labworks(what);
    }

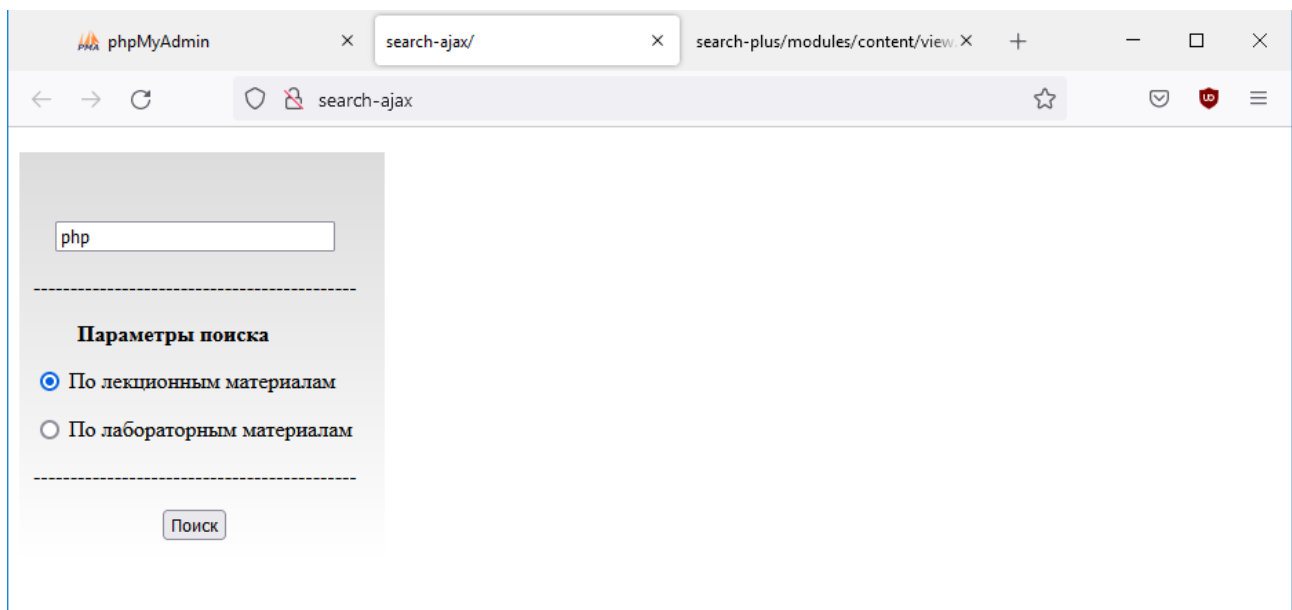
    function content_lectures (id)
    {
        $.ajax({
            dataType: 'json',
            url:
'../../modules/content/view.php?where=lectures&what='+id,
            success: function(jsondata)
            {
                $('<div>.content</div>').html('<b>Тема лекции:</b> ' +
jsondata.topic + '<br><b>Название лекции:</b> ' + jsondata.name+
'<br><b>Материалы лекции:</b> ' + jsondata.content);
            }
        });
    };

    function content_labworks (id)
    {
        $.ajax({
            dataType: 'json',
            url:
'../../modules/content/view.php?where=labworks&what='+id,
            success: function(jsondata)
            {
                $('<div>.content</div>').html('<b>Тема лабораторной работы:</b> '
+ jsondata.name + '<br><b>Материалы лабораторной работы:</b> ' +
jsondata.content);
            }
        });
    };
};
</script>

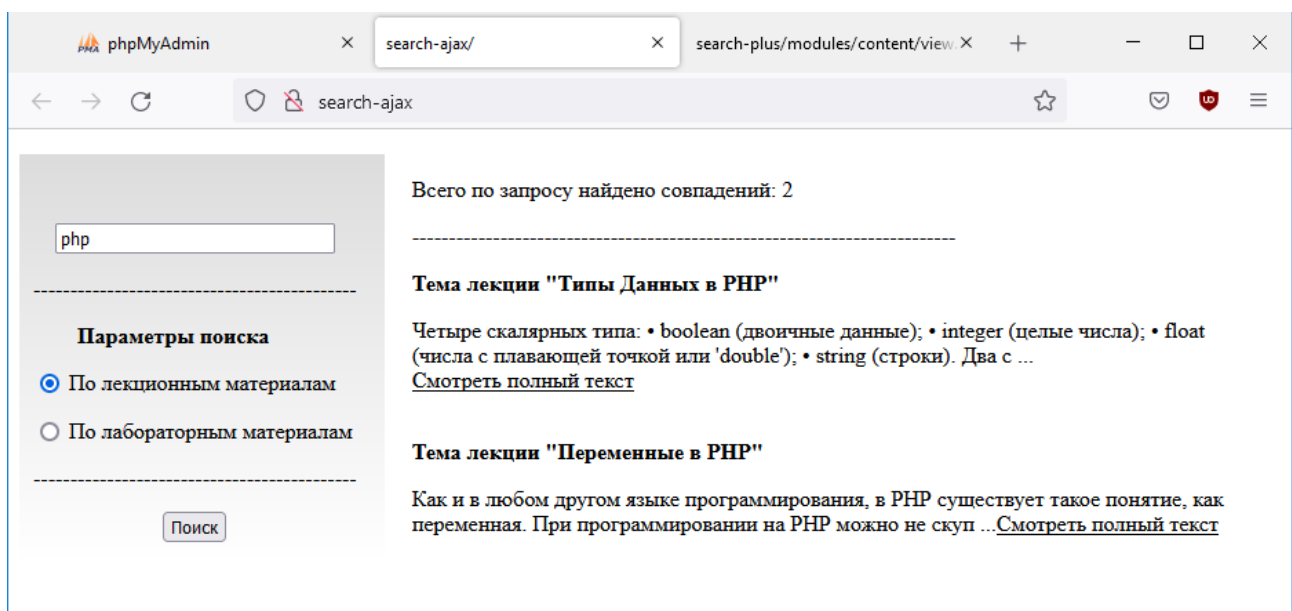
```

В итоге получим следующую последовательность отображения информации для пользователя:

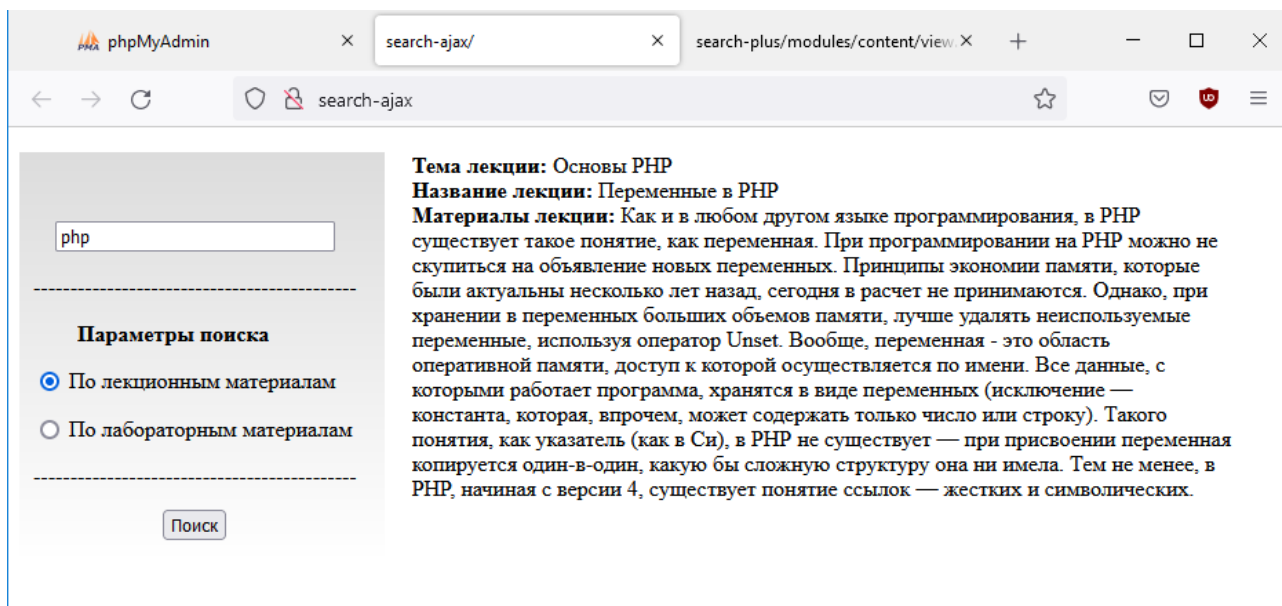
➤ Поисковая форма.



- Результаты поиска со ссылками на развернутое отображение материалов.



- Отображение материалов ресурса.



Выводы

Таким образом в данной лекции были рассмотрены:

1. Различные стратегии извлечения из БД и отображения информации при переходе пользователя по пункту меню.
2. Вариант реализации в определенной степени «безопасного» поиска с дополнительной возможностью выбора, где искать (по лекциям или лабораторным работам). Продемонстрирована реализация поиска как через перезагрузку страницы, так и с использованием асинхронных процессов.