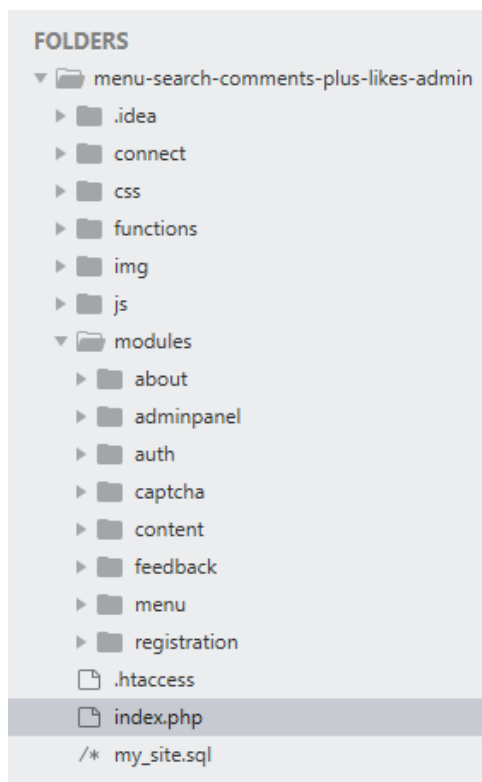


Лекция 2. Общая структура проекта «УМК». Регистрация, авторизация и реализация формы обратной связи на РНР

1. Общая структура проекта

Прежде, чем приступить к рассмотрению разработки (интеграции) отдельных модулей авторизации, регистрации и обратной связи, обобщим общую структуру проекта с учетом разработанных модулей формирования меню, отображения и поиска контента, а также планируемых модулей комментариев, лайков и административной панели.

В целом в ходе проектирования проекта «учебно-методический комплекс» выстроилась следующая структура папок.

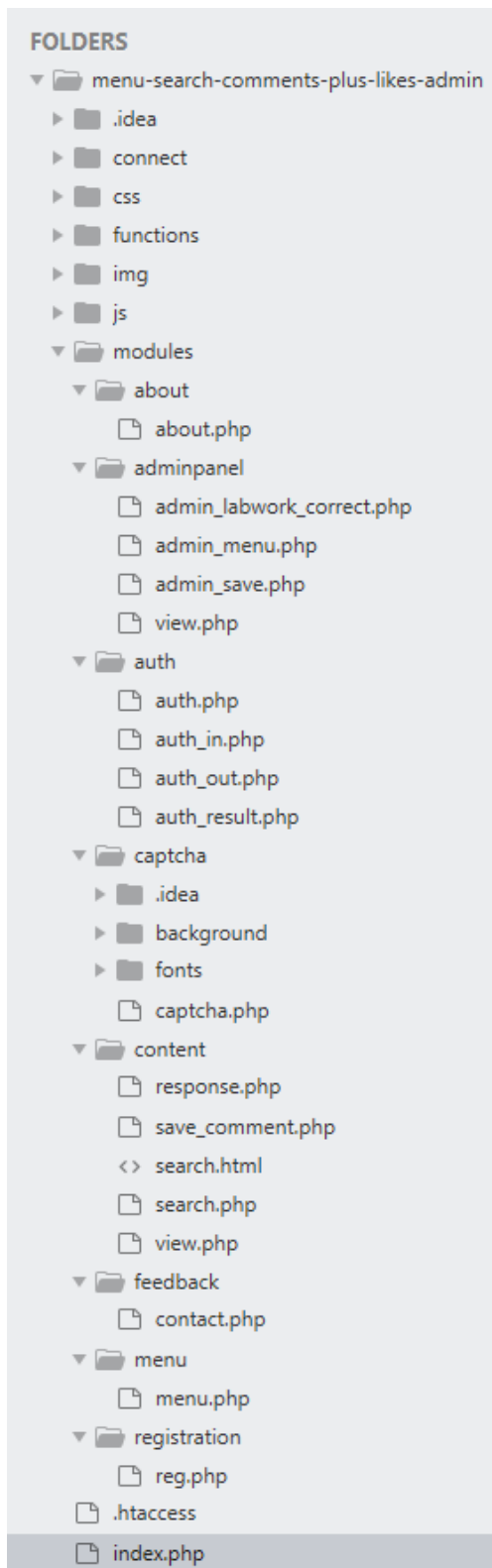


Назначение каталогов представлено в следующей таблице.

connect	Содержит файлы (файл) подключения (например, к БД)
css	Содержит файлы (файл) со стилями CSS
functions	Содержит файлы (файл) с функциями
img	Содержит картинки, использованные на сайте
js	Содержит скрипты JS
modules	Может содержать дополнительно подпапки, соответствующие, различным модуля. Например, папка captcha будет содержать все необходимые файлы, реализующие капчу, а папка auth будет содержать все необходимые файлы для входа и выхода пользователя, и т.д.
about	Содержит файлы, отображающие общую информацию о проекте
adminpanel	Содержит файлы, реализующие подсистему администрирования
auth	Содержит файлы, реализующие функцию авторизации
captcha	Содержит файлы, реализующие каптчу
content	Содержит файлы, реализующие отображение учебных материалов на страницах ресурса и описк по ним
feedback	Содержит файлы, реализующие функцию обратной связи
menu	Содержит файлы, реализующие меню
registration	Содержит файлы, реализующие функцию регистрации

Отметим, что в корневой папке проекта обычно хранятся конфигурационные файлы, файлы с разметкой и стартовый файл `index.php`.

На следующем сриншоте представлена структура файлов и папок предлагаемого к рассмотрению проекта “Учебно-методический комплекс”. Программная реализация модулей `content` и `menu`, а также структура необходимых таблиц в БД была рассмотрена в предыдущих лекциях. В данной лекции затронем вопросы реализации модулей `auth`, `registration` и `feedback`. А в последующих лекция будет рассмотрена стратегия дополнения модуля `content` функционалом комментирования и лайков, а также реализация модуля панели администрирования `adminpanel`.



Структура БД, необходимой для хранения всей информации данного проекта может быть следующей.

my_site user id : smallint(6) login : varchar(64) surname : varchar(64) first_name : varchar(64) second_name : varchar(64) password : varchar(64) salt : int(11) mail : varchar(64) reg_date : datetime last_act : datetime status : varchar(32)	my_site menu id : int(10) unsigned title : varchar(32) parent_id : int(10) unsigned link : varchar(128)	my_site lectures lecture_id : int(10) unsigned id : int(10) unsigned topic : varchar(256) name : varchar(256) content : text count_like : int(10) unsigned count_unlike : int(10) unsigned
my_site likes_lecture id : int(10) unsigned id_user : int(10) unsigned id_content : int(10) unsigned	my_site comment_lecture comment_id : int(10) unsigned id : int(10) unsigned login : varchar(64) comment : text date : datetime id_reply : int(64) unsigned	my_site labworks labwork_id : int(10) unsigned id : int(10) unsigned lecture_id : int(10) name : varchar(256) content : text count_like : int(10) unsigned count_unlike : int(10) unsigned
my_site likes_labwork id : int(10) unsigned id_user : int(10) unsigned id_content : int(10) unsigned	my_site comment_labwork comment_id : int(10) unsigned id : int(10) unsigned login : varchar(64) comment : text date : datetime id_reply : int(64) unsigned	

Как видно из рисунка, для информации о пользователях будет использоваться отдельная таблица *user*, учебные материалы в виде лекций и лабораторных работ будут храниться в таблицах *lectures* и *labworks*, для хранения комментариев также создаются две таблицы – *comment_lecture* и *comment_labwork*. А вот для реализации лайков часть информации (вспомогательной, чтобы на лету не делать постоянные пересчеты) будет храниться в таблицах *likes_lecture* и *likes_labwork*, при этом для фиксации, например, кто поставил лайк (по идентификатору пользователя), будут использованы опять же две отдельные таблицы: *likes_lecture* и *likes_labwork*.

Теперь начнем рассматривать реализацию отдельных функциональных модулей либо отдельных частей проекта «УМК». Первыми из них станут следующие:

1. Стартовая страница (index.php);
2. Модуль регистрации;
3. Модуль авторизации;
4. Модуль обратной связи.

2. Стартовая страница index.php с основным меню

Начнем со структуры стартовой страницы. Общий ее вид будет следующим.

МЕНЮ САЙТА

Лекции

[Лекция 1](#)[Лекция 2](#)[Лекция 3](#)[Лекция 4](#)

Лабораторные работы

[Лабораторная работа 1](#)[Лабораторная работа 2](#)[Самостоятельная работа](#)

Верстка стартовой страницы построена на элементах div, но для загрузки форм регистрации и авторизации, результатов валидации в один из div вставлен плавающий фрейм (iframe).

```
<?php
    if(session_status() != PHP_SESSION_ACTIVE) session_start();
    include './functions/func.php';
?>
<html>
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
<div class="header">
    <div class='header_main'>
        <a style="text-decoration: none; color: gray"
href='index.php'>УМК "Веб-программирование электронных изданий"</a>
    </div>
    <div class='header_menu'>
        <?php
            echo "<li class=\"menu2\" style=\"list-style: none; display:
inline; margin: 10px\"><a style=\"text-decoration: none\"
href=\"../modules/about/about.php\" target=\"CONTENT\">О
проекте</a></li>";
            echo "<li class=\"menu2\" style=\"list-style: none; display:
inline; margin: 10px\"><a style=\"text-decoration: none\"
href=\"../modules/feedback/contact.php\" target=\"CONTENT\">Обратная
связь</a></li>&emsp;&emsp;&emsp;&emsp;&emsp;";
            if ($_SESSION['login'] == '') {
```

```

        echo "<li class=\"menu2\" style=\"list-style: none;
display: inline; margin: 10px\"><a style=\"text-decoration: none\"
href=\"./modules/auth/auth.html\" target=\"CONTENT\">Войти</a></li>";
    }
    else{
        $login=$_SESSION['login'];
        if (isset($_GET['IsAdminPanel']))
$IsAdminPanel=$_GET['IsAdminPanel']; else unset($IsAdminPanel);
        if (IsAdmin($login)==True)
        {
            echo "<li class=\"menu2\" style=\"list-style:
none; display: inline; margin: 8px\"><a style=\"text-decoration: none\"
href=\"./modules/auth/auth_out.php\" target=\"CONTENT\">Выйти
($login)</a></li>";

            echo "<li class=\"menu2\" style=\"list-style:
none; display: inline; margin: 8px\"><a style=\"text-decoration: none\"
href=\"index.php?IsAdminPanel=true\">Админпанель</a></li>";
        }
        else
        {
            echo "<li class=\"menu2\" style=\"list-style:
none; display: inline; margin: 8px\"><a style=\"text-decoration: none\"
href=\"./modules/auth/auth_out.php\" target=\"CONTENT\">Выйти
($login)</a></li>";
        }
    }
    ?>
    <li class="menu2" style="list-style: none; display: inline;
margin: 10px"><a style="text-decoration: none"
href='./modules/registration/reg.php'
target="CONTENT">Регистрация</a></li>
    <li class="menu2" style="list-style: none; display: inline;
margin: 10px"><a style="text-decoration: none"
href='./modules/content/search.html' target="CONTENT">Поиск</a></li>
</div>
</div>
<div class="menu">
    <?php
        if (IsAdmin($_SESSION['login'])==True &&
$IsAdminPanel==true){
            include ("modules/adminpanel/admin_menu.php");
        }
        else
        {
            include ("modules/menu/menu.php");
        }
    ?>
</div>
<div class="content">

```

```

        <iframe alloytransparency name="CONTENT"
src="./img/giphy1.gif" align="middle" width="107%" height="100%"
scrolling="no" display:block style="border: 0;">Ваш браузер не
поддерживает iframe</iframe>
    </div>
</div>
<div class="footer">
    Романенко Дмитрий, 2020
</div>
</body>
</html>

```

Отметим, что в коде используется отдельно разработанная функция, позволяющая проверять роль пользователя `IsAdmin`, программная реализация которой будет рассмотрена в соответствующей лекции. Также необходимо отметить, что разметка меню уже сейчас содержит строку, учитывающие авторизацию пользователя с ролью администратор, которому необходимо предоставить ссылку на вход в админпанель.

```

if (IsAdmin($login)==True) {
    echo "<li class=\"menu2\" style=\"list-style: none; display:
inline; margin: 8px\"><a style=\"text-decoration: none\"
href=\"./modules/auth/auth_out.php\" target=\"CONTENT\">Выйти
($login)</a></li>";
    echo "<li class=\"menu2\" style=\"list-style: none; display:
inline; margin: 8px\"><a style=\"text-decoration: none\"
href=\"index.php?IsAdminPanel=true\">Админпанель</a></li>";
}

```

Важно понимать, что в данный файл включена вся разметка (в виду ее минимального количества), хотя на практике как правило все разделяют на отдельные файлы, например, `header.php`, `footer.php` и т.д., а в файле `index.php` по сути происходит их подключение в нужной последовательности.

Важным также является до начала разметки открыть сессию, так как в случае открытия ее позже по коду могут возникнуть ошибки, связанные с невозможностью сессии и ее параметров в `cookies`. Например

```

Warning: session_start(): Cannot send session cache limiter - headers already sent (output started at C:\OpenServer\OSPanel\domains\site1\index.php:4) in
C:\OpenServer\OSPanel\domains\site1\index.php on line 21

Warning: session_start(): Cannot send session cookie - headers already sent by (output started at C:\OpenServer\OSPanel\domains\site1\index.php:4) in
C:\OpenServer\OSPanel\domains\site1\modules\auth\auth_result.php on line 2

Warning: session_start(): Cannot send session cache limiter - headers already sent (output started at C:\OpenServer\OSPanel\domains\site1\index.php:4) in
C:\OpenServer\OSPanel\domains\site1\modules\auth\auth_result.php on line 2

```

Так же отметим, что в файл `index.php` подключается файл `auth_result.php`, который фактически в зависимости от авторизован ли пользователь или нет, загружает одно из приветственных сообщений с требуемыми пунктами меню (Регистрация, Войти, Выйти). Пункты меню «О проекте» и «Обратная связь», а также «Поиск» загружаются независимо от состояния авторизации пользователя.

Файл `auth_result.php` по сути занимается вопросом проверки авторизован ли пользователь в системе (есть ли его атрибуты в сессии) и если результат успешный (пользователь авторизован), то при загрузке стартовой страницы пункт меню «Войти»

меняется на «Выйти (Имя пользователя)». Т. е. в процессе авторизации он непосредственно не участвует, а является вспомогательным для корректного формирования головного меню.

auth_result.php

```
<?php
    if(session_status() !=PHP_SESSION_ACTIVE) session_start();
?>
<style>
    li {
        list-style-type: none;
        display: inline;
        display: block;
        padding: 0px 7px;
    }
    ul {
        margin-left: 0;
        padding-left: 0;
    }
    ul#menu li{
        display:inline-block; /* расположить пункты горизонтально */
    }
</style>
<?php
    if (!empty($_SESSION['login'])) {
        echo "<ul id=\"menu\">";
        echo "<li>Добро пожаловать на сайт, </li>";
        echo
"<li>".$_SESSION['first_name']."&nbsp;".$_SESSION['second_name']."!". "<
/li>";
        echo "<li><a href=\"modules/auth/auth_out.php\"
target=\"content\">Выйти</a></li>";
    }
    else{
        echo "<ul id=\"menu\">";
        echo "<li>Добро пожаловать на сайт, ГОСТЬ!
&emsp;&emsp;</li>";
        echo "<li><a href=\"modules/registration/reg.php\"
target=\"content\">Регистрация</a></li>";
        echo "<li><a href=\"modules/auth/auth.html\"
target=\"content\">Войти</a></li>";
        echo "</ul>";
        echo "<br>";
    }
?>
```

МЕНЮ САЙТА

Лекции

[Лекция 1](#)

[Лекция 2](#)

[Лекция 3](#)

[Лекция 4](#)

Лабораторные работы

[Лабораторная работа 1](#)

[Лабораторная работа 2](#)

[Самостоятельная работа](#)

HELLO
EVERYONE

МЕНЮ САЙТА

Лекции

[Лекция 1](#)

[Лекция 2](#)

[Лекция 3](#)

[Лекция 4](#)

Лабораторные работы

[Лабораторная работа 1](#)

[Лабораторная работа 2](#)

[Самостоятельная работа](#)

HELLO
EVERYONE

3. Регистрация пользователя. Валидация полей формы

Для реализации удобной и эффективной системы регистрации необходимо разработать форму регистрации со специальными полями, в которых будут отображаться результаты валидации полей, создать соответствующую таблицу в БД, хранящую информацию о пользователях, реализовать удобную для пользователя проверку полей регистрации. В данном варианте форма для регистрации будет выглядеть следующим образом: логин, поле для ввода пароля и его подтверждения, поля для ввода электронной почты, фамилии, имени и отчества, не видимые (при начальной загрузке формы) поля для отображения информации о результатах валидации. К форме регистрации добавляется рассмотренная ранее captcha.

Регистрация

Ваш логин:

Ваш пароль:

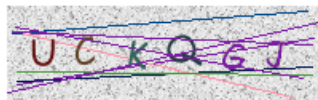
Подтверждение пароля:

Е-mail:

Фамилия:

Имя:

Отчество:



[Обновить капчу](#)

Помимо простой проверки на пустоту полей, предполагается проверка логина (разрешены исключительно латинские буквы) и e-mail с помощью регулярных выражений. Пароль в базе данных будет храниться в виде хэша, для обеспечения надежности последнего будет использовано случайное трехзначное число (salt). Также предполагается записывать в БД дату регистрации, дату последней аутентификации, и статус пользователя, который, как правило необходим при дальнейшем предоставлении прав, а также при реализации в дальнейшем системы администрирования. Отметим, что все пользователи при регистрации по умолчанию принимают статус user. В связи с этим таблица с пользователями примет следующий вид.

Структура таблицы Связи										
	#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1	id	smallint(5)		UNSIGNED	Нет	Нем		AUTO_INCREMENT	
<input type="checkbox"/>	2	login	varchar(64)	utf8_general_ci		Нет	Нем			
<input type="checkbox"/>	3	surname	varchar(64)	utf8_general_ci		Нет	Нем			
<input type="checkbox"/>	4	first_name	varchar(64)	utf8_general_ci		Нет	Нем			
<input type="checkbox"/>	5	second_name	varchar(64)	utf8_general_ci		Нет	Нем			
<input type="checkbox"/>	6	password	varchar(64)	utf8_general_ci		Нет	Нем			
<input type="checkbox"/>	7	salt	int(11)			Нет	Нем			
<input type="checkbox"/>	8	mail	varchar(64)	utf8_general_ci		Нет	Нем			
<input type="checkbox"/>	9	reg_date	datetime			Нет	Нем			
<input type="checkbox"/>	10	last_act	datetime			Нет	Нем			
<input type="checkbox"/>	11	status	varchar(32)	utf8_general_ci		Нет	user			

Необходимо обратить внимание, что даты будут записываться в формате Y-m-d H:i:s, т.е. перед записью должно быть выполнено соответствующее форматирование. Хэш пароля будет рассчитываться следующим образом: md5(md5(password).salt), т.е. пароль хэшируется с помощью алгоритма md5, далее к полученной последовательности дописывается случайное трехзначное число, а затем вся последовательность заново хэшируется. Результат будет храниться в БД.

Уведомление об ошибке в заполнении формы появится после перезагрузки страницы, формы регистрации, но при этом все введенные данные (кроме полей с паролями) должны сохраняться. Если все поля формы заполнены корректно, то информация о пользователе должна записаться в БД, об успешной регистрации будет выведено советующее сообщение, пользователь автоматически должен пройти процедуру аутентификации (т.е. его данные после регистрации записываются в сессию), а затем выполняется полная перезагрузка стартовой страницы для применения изменений об аутентифицированном пользователе.

В связи с тем с тем, что результаты валидации предполагается записывать в специальные поля на форме, целесообразно файл формы и файл, обрабатывающий форму по нажатию на кнопку объединить вместе, т.е. фактически файл будет вызывать самого себя. Назовем его reg.php, полный листинг которого будет следующим.

Reg.php

```
<?php
if(session_status() != PHP_SESSION_ACTIVE) session_start();
include '../../functions/func.php';

if (isset($_POST['login'])) { $login = $_POST['login']; if ($login ==
'' ) { unset($login);} } //заносим введенный пользователем логин в
переменную $login, если он пустой, то уничтожаем переменную
if (isset($_POST['password'])) { $password=$_POST['password']; if
($password == '') { unset($password);} } //заносим введенный
пользователем пароль в переменную $password, если он пустой, то
уничтожаем переменную
if (isset($_POST['password2'])) { $password2=$_POST['password2']; if
($password2 == '') { unset($password2);} }
if (isset($_POST['email'])) { $email=$_POST['email']; if ($email == '')
{ unset($email);} }
if (isset($_POST['surname'])) { $surname = $_POST['surname']; if
($surname == '' ) { unset($surname);} }
if (isset($_POST['first_name'])) { $first_name = $_POST['first_name'];
if ($first_name == '' ) { unset($first_name);} }
if (isset($_POST['second_name'])) { $second_name =
$_POST['second_name']; if ($second_name == '' )
{ unset($second_name);} }

if (check_captcha() == true)
{
    if(isset($_POST["go"])):
        $el=null;
        $login=trim($_POST["login"]);
        $login=strip_tags($login);
        $login=stripslashes($login);
        if(strlen($login) == "0"):
            $el="Заполните поле 'Ваш логин'<br>";
        endif;
        if (!preg_match('/^([a-zA-Z0-9])(\w|-|_)+([a-z0-9])$/is',
$login)):
            $el="Введенный логин не соответствует требованиям<br>";
```

```

endif;

include '../../connect/bd.php'; //подключаемся к БД
$query="SELECT id FROM user WHERE login='$login'";
$result = mysqli_query($link, $query) or die("Ошибка выполнения
запроса" . mysqli_error($link));
if ($result){
    $row = mysqli_fetch_row($result);
    if (!empty($row[0])) $e1="Данный логин занят"; // проверка
на существование в БД такого же логина
}
$e2=null;
$password=trim($_POST["password"]);
$password=strip_tags($password);
$password=htmlspecialchars($password,ENT_QUOTES);
$password=stripslashes($password);
if(strlen($password)==="0"):
    $e2.="Заполните поле 'Пароль'<br>";
endif;

$e3=null;
$password2=trim($_POST["password2"]);
$password2=strip_tags($password2);
$password2=htmlspecialchars($password2,ENT_QUOTES);
$password2=stripslashes($password2);
if(strlen($password2)==="0"):
    $e3.="Заполните поле 'Подтверждение пароля'<br>";
endif;

if ($password!=$password2):
    $e2.="Пароли не совпадают<br>";
endif;

$e4=null;
$email=trim($_POST["email"]);
$email=strip_tags($email);
$email=htmlspecialchars($email,ENT_QUOTES);
$email=stripslashes($email);
if(!filter_var($email, FILTER_VALIDATE_EMAIL)):
    $e4.="Неверный e-mail<br>";
endif;

$eEn=$e1.$e2.$e3.$e4;
if($eEn==""):
    $salt = mt_rand(100, 999);
    $tm = date("Y-m-d H:i:s",time());
    $password = md5(md5($password).$salt);
    $query="INSERT INTO user (login, password, salt, mail,
reg_date, last_act, surname, first_name, second_name) VALUES

```

```

('$login','$password','$salt','$email','$tm','$tm','$surname','$first_n
ame','$second_name');"
        $result=mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
        if ($result) //пишем данные в БД и авторизовываем
пользователя
        {
            $query="SELECT * FROM user WHERE login='$login'";
            $rez = mysqli_query($link, $query);
            if ($rez)
            {
                echo "записываем в сессию";
                $row = mysqli_fetch_assoc($rez);
                $_SESSION['id'] = $row['id'];
                $_SESSION['login'] = $row['login'];
                $_SESSION['surname']=$row['surname'];
                $_SESSION['first_name']=$row['first_name'];
                $_SESSION['second_name']=$row['second_name'];
                $_SESSION['status']=$row['status'];
                //echo "Вы успешно зарегистрированы,
".$_SESSION['login'];
                mysqli_close($link);
                // выводим уведомление об успехе операции и
перезагружаем страничку
                print "<script language='Javascript'
type='text/javascript'>
                    alert ('Вы успешно зарегистрировались и
авторизовались! Спасибо!');
                    function reload(){top.location =
'../../index.php'};
                    setTimeout('reload()', 0);
                </script>";
            }
            else
            {
                print "<script language='Javascript'
type='text/javascript'>
                    alert ('Ваши данные не были внесены в БД!');
                </script>";
            }
        }
    endif;
endif;
}
?>
<html>
<body>

<div class="reg">

```

```
<h2>Регистрация</h2>
<form action="/modules/registration/reg.php" method="post">
  <p>
    <label>Ваш
логин:&emsp;&emsp;&emsp;&emsp;&emsp;&nbsp;</label>
    <input type="text" name="login" value="<?=@$login;?>"
size="25" maxlength="25" autofocus>
    <span class="error"><?=@$e1;?></span>
  </p>
  <p>
    <label>Ваш
пароль:&emsp;&emsp;&emsp;&emsp;&emsp;&nbsp;&nbsp;&nbsp;</label>
    <input name="password" type="password" size="25"
maxlength="25">
    <span class="error"><?=@$e2;?></span>
  </p>
  <p>
    <label>Подтверждение пароля:</label>
    <input name="password2" type="password" size="25"
maxlength="25">
    <span class="error"><?=@$e3;?></span>
  </p>
  <p>
    <label>E-
mail:&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&nbsp;</label>
    <input name="email" value="<?=@$email;?>" type="text"
size="25" maxlength="25">
    <span class="error"><?=@$e4;?></span>
  </p>
  <p>
    <label>Фамилия:&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;</label>
    <input name="surname" value="<?=@$surname;?>"
type="text" size="25" maxlength="25">
    <span class="error"><?=@$e5;?></span>
  </p>
  <p>
    <label>Имя:&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&nbsp;</label>
    <input name="first_name" value="<?=@$first_name;?>"
type="text" size="25" maxlength="25">
    <span class="error"><?=@$e6;?></span>
  </p>
  <p>
    <label>Отчество:&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;</label>
    <input name="second_name" value="<?=@$second_name;?>"
type="text" size="25" maxlength="25">
    <span class="error"><?=@$e7;?></span>
  </p>
</form>
```

```

        </p>
        <p><input type="hidden" name="go" value="5"></p>

        ... Здесь должны быть строки скрипта вашей каптчи ...

        <p style="margin-left: 15%">
            <input type="submit" name="submit"
value="Зарегистрироваться">
        </p>
    </form>
</div>
</body>
</html>

```

Необходимо отметить, что листинг данного скрипта взят из готового проекта, а поэтому путь к файлу, выполняющему подключение к БД, задан с учетом структуры папок всего проекта.

Рассмотрим некоторые его части более подробно. Данный файл фактически состоит из двух частей: строки, отвечающие за обработку формы (отображены желтым цветом) и строки кода, отвечающие за отображение элементов формы, включая результаты валидации и сохранение ранее введенных пользователем данных (отображены зеленым цветом).

Что касается формы, то основным моментам нужно отнести использование тегов `` для отображения результатов валидации (при начальной загрузке формы, как и положено, в них ничего нет, а какая-либо информация появляется при необходимости после обработки формы).

```
<span class="error"><?=@$e1;?></span>
```

Причем для каждого поля зарезервирована отдельная переменная, в которую будут записана информация о результатах валидации (в примере выше `$e1`, обрамленная тегами `php`-скрипта для непосредственного вывода ошибок при валидации поля `login`).

Аналогичным образом осуществляется запоминание введенных данных в поля формы при наличии ошибок валидации.

```
<input type="text" name="login" value="<?=@$login;?>" size="25"
maxlength="25" autofocus>
```

```
<input name="email" value="<?=@$email;?>" type="text" size="25"
maxlength="25">
```

```
<input name="surname" value="<?=@$surname;?>" type="text" size="25"
maxlength="25">
```

```
<input name="first_name" value="<?=@$first_name;?>" type="text"
size="25" maxlength="25">
```

```
input name="second_name" value="<?=@$second_name;?>" type="text"
size="25" maxlength="25">
```

Также необходимо отметить, что на форме размещен скрытый элемент

```
<input type="hidden" name="go" value="5">,
```

который при обработке будет использоваться как критерий, были ли данные с формы отправлены. Проверяется факт отправки данных с формы следующим образом

```
if(isset($_POST["go"]))
{
}
```

Валидацию будем выполнять поэтапно. Так, например, данные, введенные в поле login, проверим следующим образом.

```
$el=null;
$login=trim($_POST["login"]);
$login=strip_tags($login); // вырезаем теги
$login=stripslashes($login);
if(strlen($login)==0):
    $el.="Заполните поле 'Ваш логин'<br>";
endif;
if (!preg_match('/^([a-zA-Z0-9])(\w|-|_)+([a-z0-9])$/is',
$login)):
    $el.="Введенный логин не соответствует требованиям<br>";
endif;
```

Как видно, в результате проверки переменная el будет равна null (или ''), если все правила соблюдены, либо в нее будет записана соответствующая текстовая информация. В данном примере данные с поля login обрабатываются специальными функциями, чтобы минимизировать возможности атак, проверяется, чтобы не было пустым, далее с использованием регулярных выражений проверяем, чтобы логин состоял из латинских символов, цифр и некоторых служебных символов. Также необходимо проверить, чтобы введенный логин был свободен – сделать это можно следующими строками кода.

```
include '../../connect/bd.php'; //подключаемся к БД
$query="SELECT id FROM user WHERE login='$login'";
$result = mysqli_query($link, $query) or die("Ошибка выполнения
запроса" . mysqli_error($link));
if ($result){
    $row = mysqli_fetch_row($result);
    if (!empty($row[0])) $el.="Данный логин занят"; // проверка
на существование в БД такого же логина
}
```

Т.е. выполняется подключение к БД (если файл подключения к БД лежит в одной папке с файлом reg.php, то строка может выглядеть следующим образом.

```
include ('bd.php');
```

Далее отправляется select запрос на предмет поиска, например, id пользователя с логином, совпадающим с логином пользователя.

Очевидно, что аналогичным образом можно проверять поля с вводимыми паролями, email и т.д. Также можно все проверки вынести в отдельную функцию и разместить ее в соответствующем файле.

В итоге все переменные, хранящие результаты проверки (в данном примере это e1, e2, e3, e4) объединяются с помощью операции конкатенации. Если результатом такой операции будет null, значит ошибок не было.

```
$eEn=$e1.$e2.$e3.$e4;
        if($eEn==null)
        {
// добавляем данные в БД и проходим авторизацию
// выводим сообщение об успешной регистрации и перезагружаем страницу
        }
    Else{
//выводим сообщение об ошибке
    }
```

Сообщения выводятся и перезагружается страница с помощью JS.

```
print "<script language='Javascript' type='text/javascript'>
    <!--
    alert ('Вы успешно зарегистрировались и
авторизировались! Спасибо!');
    function reload(){top.location =
'../../index.php'};
    setTimeout('reload()', 0);
    -->
</script>";
```

```
print "<script language='Javascript' type='text/javascript'>
      alert ('Ваши данные не были снесены в БД!');
    </script>";
```

В итоге получим следующее функционирование обновленной системы регистрации. Отметим, что для решения вопросов взаимного расположения элементов, форм, цветов и типов шрифтом и т.д. может и должно применяться CSS.

[illegible]

МЕНЮ САЙТА

Лекции

[Лекция 1](#)[Лекция 2](#)[Лекция 3](#)[Лекция 4](#)

Лабораторные работы

[Лабораторная работа 1](#)[Лабораторная работа 2](#)[Самостоятельная работа](#)

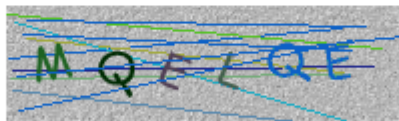
Регистрация

Ваш логин:	<input type="text" value="Peter"/>
Ваш пароль:	<input type="password" value="•"/>
Подтверждение пароля:	<input type="password" value="•"/>
E-mail:	<input type="text" value="peter@tut.by"/>
Фамилия:	<input type="text" value="Петров"/>
Имя:	<input type="text" value="Петр"/>
Отчество:	<input type="text" value="Петрович"/>

[Обновить капчу](#)

Регистрация

Ваш логин:	<input type="text" value="Peter"/>
Ваш пароль:	<input type="password" value="•"/>
Подтверждение пароля:	<input type="password" value="•"/>
Е-mail:	<input type="text" value="peter@google.com"/>
Фамилия:	<input type="text" value="Петров"/>
Имя:	<input type="text" value="Петр"/>
Отчество:	<input type="text" value="Петрович"/>



[Обновить капчу](#)

Зарегистрироваться

УМК "Веб-программирование электронных изданий"

[О проекте](#) [Обратная связь](#)

[Выйти \(Dmitri\)](#) [Регистрация](#) [Поиск](#)

МЕНЮ САЙТА

Лекции

- [Лекция 1](#)
- [Лекция 2](#)
- [Лекция 3](#)
- [Лекция 4](#)

Лабораторные работы

- [Лабораторная работа 1](#)
- [Лабораторная работа 2](#)

Самостоятельная работа

вносим в сессию

⊕ menu-search-comments-plus-likes-admin

Вы успешно зарегистрировались и авторизовались! Спасибо!

ОК

МЕНЮ САЙТА

Лекции

[Лекция 1](#)

[Лекция 2](#)

[Лекция 3](#)

[Лекция 4](#)

Лабораторные работы

[Лабораторная работа 1](#)

[Лабораторная работа 2](#)

[Самостоятельная работа](#)

HELLO
EVERYONE

Регистрация и автоматически авторизация прошли успешно. Однако, если допустить ошибки, например, ввести уже существующий логин, или ввести различные пароли, или не правильный e-mail, то на форме регистрации будут выведены соответствующие сообщения об ошибках.

Регистрация

Ваш логин: Данный логин занят

Ваш пароль: Пароли не совпадают

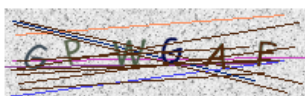
Подтверждение пароля:

Е-mail:

Фамилия:

Имя:

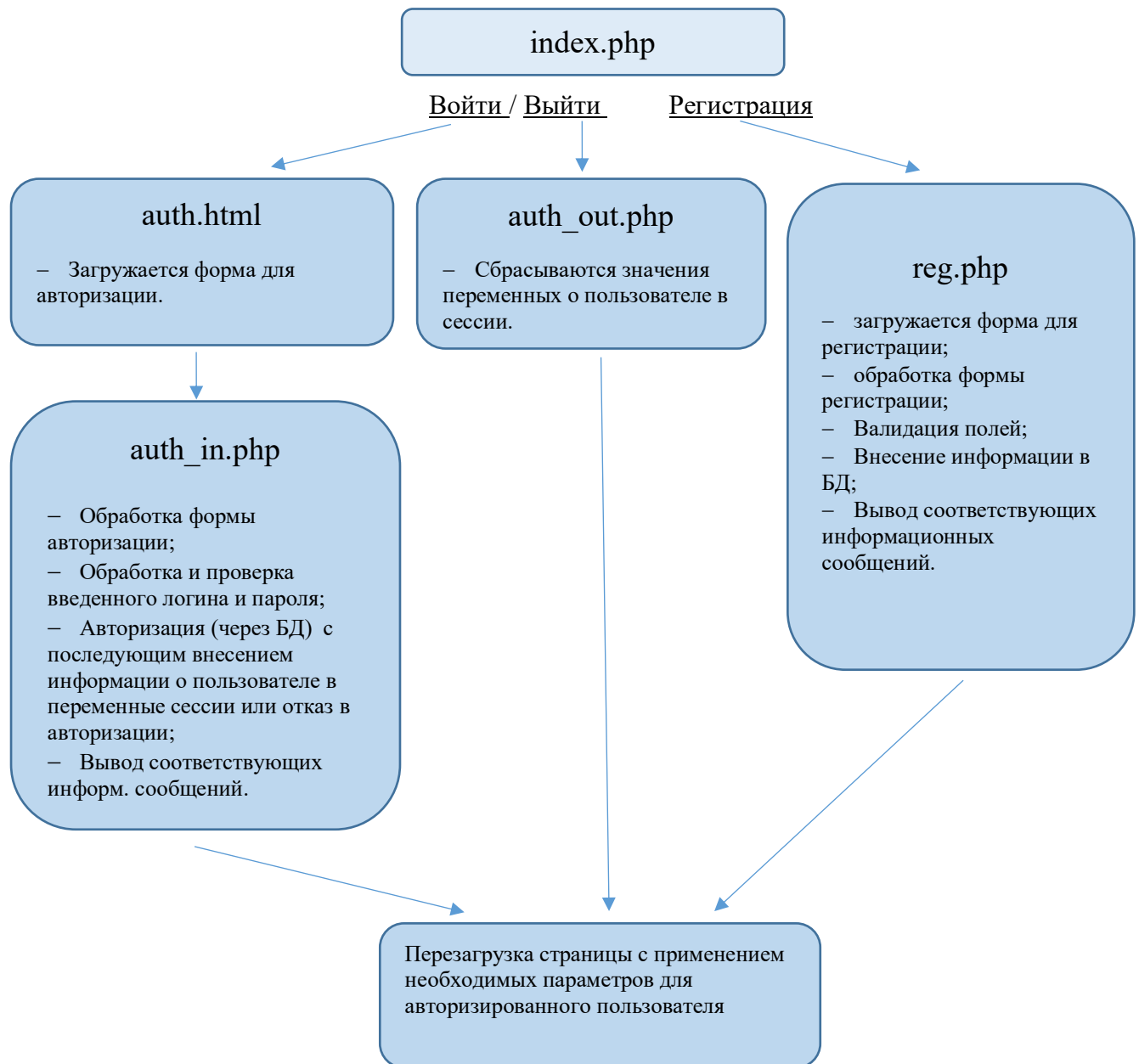
Отчество:



[Обновить капчу](#)

4. Авторизация пользователя. Выход пользователя

Для подсистемы авторизации пользователя (вход/выход) необходимы два файла, отвечающие за вход и выход пользователя (auth_in.php, auth_out.php), а также файл с формой авторизации auth.php.



Реализация пункта меню *Войти* будет выполнена через загрузку формы авторизации (файл auth.php) с последующей обработкой введенных пользователем логина и пароля в файле auth_in.php, который фактически при корректных введенных данных «запишет» соответствующую информацию о пользователе в сессию.

Отметим, что форма будет загружаться из php, а не html файла, так в случае ввода пользователем неправильного пароля, логин должен быть записан в сессию и при перезагрузке формы авторизации «вставлен» в соответствующее поле формы.

auth.php

```

<?php
    if(session_status() !=PHP_SESSION_ACTIVE) session_start();
?>
<html>
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
</head>
<body>
<div class="auth">
    <form action="/modules/auth/auth_in.php" method="post">
        <p style="margin-left: 20%">
            <br>
            <br>
            <label>Логин:&nbsp;   &nbsp;  </label>
            <input name="login" value="<?=@$_SESSION['temp_login'];?>"
type="text" size="28" maxlength="28">
        </p>
        <p style="margin-left: 20%">
            <label>Пароль:</label>
            <input name="password" type="password" size="28"
maxlength="28">
        </p>
        <p style="margin-left: 35%">
            <br>
            <input type="submit" name="submit" value="Войти"
style="width:100px">
        </p>
    </form>
</div>
</body>
</html>

```

auth_in.php

```

<?php
    if(session_status() !=PHP_SESSION_ACTIVE) session_start();
?>
<?php
    if (isset($_POST['login'])) { $login = $_POST['login']; if ($login
== '') { unset($login);} }
    if (isset($_POST['password'])) { $password=$_POST['password']; if
($password == '') { unset($password);} }
    if (empty($login) or empty($password)) {
        print "<script language='Javascript' type='text/javascript'>
            alert ('Вы заполнили не все поля!');
            function reload(){location =
'../../modules/auth/auth.html'};
            setTimeout('reload()', 0);

```

```

        </script>";
        exit();
    }

    $login = stripslashes($login);
    $login = htmlspecialchars($login);
    $password = stripslashes($password);
    $password = htmlspecialchars($password);
    $login = trim($login);
    $password = trim($password);
    $_SESSION['temp_login']=$login;

    include '../connect/bd.php';
    $query ="SELECT * FROM user WHERE login='$login'";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    $row = mysqli_fetch_assoc($result);

    if (empty($row['id']))
    {

        mysqli_close($link);
        print "<script language='Javascript' type='text/javascript'>
            alert ('Такого логина не существует!');
            function reload(){top.location = 'auth.php'};
            setTimeout('reload()', 0)
        </script>";
    }
    else
    {
        if ($row['password']==md5(md5($password).$row['salt']))
        {

            $_SESSION['login']=$row['login'];
            $_SESSION['surname']=$row['surname'];
            $_SESSION['first_name']=$row['first_name'];
            $_SESSION['second_name']=$row['second_name'];
            $_SESSION['id_user']=$row['id'];
            $_SESSION['status']=$row['status'];
            print "<script language='Javascript'
type='text/javascript'>
                function reload(){top.location = '../index.php'};
                setTimeout('reload()', 0)
            </script>";

        }
        else
        {

```

```
        print "<script language='Javascript'
type='text/javascript'>
            alert ('Вы ввели не правильный пароль!');
            function reload(){location = 'auth.php'};
            setTimeout('reload()', 0)
        </script>";
    }
    mysqli_close($link);
}

?>
```

УМК "Веб-программирование электронных изданий"

[О проекте](#)[Обратная связь](#)[Войти](#)[Регистрация](#)[Поиск](#)

МЕНЮ САЙТА

Лекции

[Лекция 1](#)[Лекция 2](#)[Лекция 3](#)[Лабораторная работа 1](#)[Лабораторная работа 2](#)[Самостоятельная работа](#)Логин: Пароль:

УМК "Веб-программирование электронных изданий"

[О проекте](#)[Обратная связь](#)[Войти](#)[Регистрация](#)[Поиск](#)

МЕНЮ САЙТА

Лекции

[Лекция 1](#)[Лекция 2](#)[Лекция 3](#)[Лабораторная работа 1](#)[Лабораторная работа 2](#)[Самостоятельная работа](#)Логин: Пароль:

МЕНЮ САЙТА

Лекции

[Лекция 1](#)

[Лекция 2](#)

[Лекция 3](#)

[Лекция 4](#)

Лабораторные работы

[Лабораторная работа 1](#)

[Лабораторная работа 2](#)

[Самостоятельная работа](#)

HELLO
EVERYONE

Для реализации пункта меню *Выйти* будет использован файл `auth_out.php`, который фактически «обнулит» (сбросит) соответствующую информацию о пользователе в сессии.

`auth_out.php`

```
<?php
    if(session_status() != PHP_SESSION_ACTIVE) session_start();
?>
<?php
    $_SESSION['login']='';
    $_SESSION['id']='';
    $_SESSION['surname']='';
    $_SESSION['first_name']='';
    $_SESSION['second_name']='';
    $_SESSION['status']='';

    print "<script language='Javascript' type='text/javascript'>
        function reload(){top.location = '../..'/index.php'};
        setTimeout('reload()', 0);
    </script>";
?>
```

В результате выполнения скрипта возвращаемся к исходному (начальному) состоянию проекта.

МЕНЮ САЙТА

Лекции

[Лекция 1](#)

[Лекция 2](#)

[Лекция 3](#)

[Лекция 4](#)

Лабораторные работы

[Лабораторная работа 1](#)

[Лабораторная работа 2](#)

[Самостоятельная работа](#)



5. Обратная связь

Форма для обратной связи - незаменимая вещь, так как с ее помощью посетители всегда могут связаться с владельцем веб-ресурса. Для ее реализации понадобится всего одна страница для формы, на которой будет находиться разметка формы, PHP для обработки и jQuery для проверки правильности введения данных в форму.

Главную страницу создадим под названием **contact.php**. Важно, чтобы расширение этого файла было .php, так в будет содержаться не только элементы формы, но и php код для ее обработки – в таком случае не придется создавать несколько страниц, а будет достаточно всего одной.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ru" lang="ru">

<head>
    <title>Обратная связь</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"
/>
    <meta http-equiv="Content-Style-Type" content="text/css" />

<script src="../../js/jquery.min.js" type="text/javascript"></script>
<script src="../../js/jquery.validate.pack.js"
type="text/javascript"></script>

<style type="text/css">
...
</style>
</head>

<body>
```

```

<div id="contact-wrapper">
<h2 align="center">Форма обратной связи</h2>
<fieldset>
<!--Тег задает заголовок для групповых элементов-->
<legend>Оставьте сообщение:</legend>
<form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>"
id="contactform">
    <div>
        <label for="name"><strong>Имя:</strong></label>
        <input type="text" size="50" name="contactname"
id="contactname" value="" class="required" />
    </div>

    <div>
        <label for="email"><strong>Email:</strong></label>
        <input type="text" size="50" name="email" id="email"
value="" class="required email" />
    </div>

    <div>
        <label for="subject"><strong>Тема:</strong></label>
        <input type="text" size="50" name="subject"
id="subject" value="" class="required" />
    </div>

    <div>
        <label
for="message"><strong>Сообщение:</strong></label>
        <textarea rows="5" cols="50" name="message"
id="message" class="required"></textarea>
    </div>
    <input type="submit" value="Отправить" name="submit" />
</form>
</fieldset>
</div>
</body>
</html>

```

Чтобы не загромождать предыдущий код, настройки стилей приведем в отдельном листинге.

```

<style type="text/css">
body {

}
#contact-wrapper {
    width:450px;
    border:1px solid #e2e2e2;
    background:#f1f1f1;

```

```

padding:20px;
margin-left: 20%;
}
#contact-wrapper div {
clear:both;
margin:1em 0;
}
#contact-wrapper label {
display:block;
float:none;
font-size:16px;
width:auto;
}
form#contactform input {
border-color:#B7B7B7 #E8E8E8 #E8E8E8 #B7B7B7;
border-style:solid;
border-width:1px;
padding:5px;
font-size:16px;
color:#333;
}
form#contactform textarea {
font-family:Arial, Tahoma, Helvetica, sans-serif;
font-size:100%;
padding:0.6em 0.5em 0.7em;
border-color:#B7B7B7 #E8E8E8 #E8E8E8 #B7B7B7;
border-style:solid;
border-width:1px;
}
</style>

```

В самом начале был подключен фреймворк jQuery. Кроме этого нам необходимо еще подгрузить специальный плагин для проверки введенных данных и одну небольшую функцию. Все это выглядит так:

```

<script src="../../js/jquery.min.js" type="text/javascript"></script>
<script src="../../js/jquery.validate.pack.js"
type="text/javascript"></script>

<script type="text/javascript">
$(document).ready(function(){
    $("#contactform").validate();
});
</script>

```

После этого нам необходимо проверить, чтобы для обязательных полей был добавлен **class="required"**. В случае с полем для ввода Email нам необходимо добавить **class="required email"**.

В итоге должно получиться следующее.

Форма обратной связи

Оставьте сообщение:

Имя:

Email:

Тема:

Сообщение:

Отправить

МЕНЮ САЙТА

[Лекции](#)
[Лекция 1](#)
[Лекция 2](#)
[Лекция 3](#)
[Лекция 4](#)
[Лабораторные работы](#)
[Лабораторная работа 1](#)
[Лабораторная работа 2](#)
[Самостоятельная работа](#)

Форма обратной связи

Оставьте сообщение:

Имя:

Email:

Тема:

Сообщение:

Отправить

Результаты валидации полей формы с помощью JS представлены на следующем рисунке.

Форма обратной связи

Оставьте сообщение:

Имя:

Email:

Please enter a valid email address.

Тема:

Сообщение:

Далее рассмотрим обработку формы. Для этого необходимо поместить следующий код как можно выше в документе.

```
<?php
//Если форма отправлена
if(isset($_POST['submit'])) {

    //Проверка Поля ИМЯ
    if(trim($_POST['contactname']) == '') {
        $hasError = true;
    } else {
        $name = trim($_POST['contactname']);
    }
    //Проверка поля ТЕМА
    if(trim($_POST['subject']) == '') {
        $hasError = true;
    } else {
        $subject = trim($_POST['subject']);
    }
    //Проверка правильности ввода EMAIL
    if(trim($_POST['email']) == '') {
        $hasError = true;
    } else if (!preg_match('/^((([0-9A-Za-z]{1}[-0-9A-z\.\.]{1,}[0-9A-Za-z]{1})|([0-9A-Яа-я]{1}[-0-9A-я\.\.]{1,}[0-9A-Яа-я]{1}))@([-A-Za-z]{1,}\.\.){1,2}[-A-Za-z]{2,})$/u', trim($_POST['email']))) {
        $hasError = true;
    } else {
        $email = trim($_POST['email']);
    }
    //Проверка наличия ТЕКСТА сообщения
```

```

        if(trim($_POST['message']) == '') {
            $hasError = true;
        } else {
            if(function_exists('stripslashes')) {
                $comments = stripslashes(trim($_POST['message']));
            } else {
                $comments = trim($_POST['message']);
            }
        }
        //Если ошибок нет, отправить email
        if(!isset($hasError)) {
            $emailTo = 'romanenko_dmitri@icloud.com'; //Сюда введите Ваш
email
            $body = "Name: $name \n\nEmail: $email \n\nSubject: $subject
\n\nComments:\n $comments";
            $headers = 'From: My Site <'.$emailTo.>' . "\r\n" . 'Reply-
To: ' . $email;
            mail($emailTo, $subject, $body, $headers);
            $emailSent = true;
        }
    }
}
?>

```

При помощи него мы фактически повторно проверяем все поля формы, но теперь с помощью PHP. Причина для этого заключается в том, что даже если у посетителя будет отключен Javascript, он все равно сможет воспользоваться формой, которая будет проверена средствами PHP, т.е. в данном случае реализуется подход, представленный при разработке регистрации.

УМК "Веб-программирование электронных изданий"

[О проекте](#) [Обратная связь](#)

[Войти](#) [Регистрация](#) [Поиск](#)

МЕНЮ САЙТА

[Лекции](#)

[Лекция 1](#)

[Лекция 2](#)

[Лекция 3](#)

[Лекция 4](#)

[Лабораторные работы](#)

[Лабораторная работа 1](#)

[Лабораторная работа 2](#)

[Самостоятельная работа](#)

Форма обратной связи

Оставьте сообщение:

Имя:

Дмитрий

Email:

rdm@tut.by

Тема:

Привет

Сообщение:

Привет

После успешно пройденной валидации будет отправлено письмо на email, указанный в коде, и появится соответствующее окно

