

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Магнитогорский государственный технический университет им. Г.И. Носова»  
Многопрофильный колледж



**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ  
ПРАКТИЧЕСКИХ ЗАНЯТИЙ**

ПМ.02. Разработка и администрирование баз данных  
МДК.02.01.02 Программное обеспечение компьютерных сетей

для студентов специальности  
09.02.03 Программирование в компьютерных системах  
базовой подготовки

Магнитогорск, 2017

**ОДОБРЕНО:**

Предметно-цикловой комиссией Информатика и вычислительная техника  
Председатель И.Г.Зорина  
Протокол № 7 от 14 марта 2017

Методической комиссией МпК  
Протокол №4 от «23» марта 2017г

**Составитель:**

преподаватель ФГБОУ  
ВО МГТУ МпК Марина Николаевна Корчагина/

Методические указания по выполнению практических занятий разработаны на основе рабочей программы ПМ.02. Разработка и администрирование баз данных.

Содержание практических работ ориентировано на формирование общих и профессиональных компетенций по программе подготовки специалистов среднего звена по специальности 09.02.03 Программирование в компьютерных системах: МДК.02.01.02 Программное обеспечение компьютерных сетей.

## СОДЕРЖАНИЕ

1 Введение	4
2 Методические указания	5
Практическая работа 1	5
Практическая работа 2	14
Практическая работа 3	15
Практическая работа 4	17
Практическая работа 5	19
Практическая работа 6	20
Практическая работа 7	22
Практическая работа 8	23
Практическая работа 9	24
Практическая работа 10	25
Практическая работа 11	26
Практическая работа 12	27
Практическая работа 13	29
Практическая работа 14	30
Практическая работа 15	32

## 1 ВВЕДЕНИЕ

Важную часть теоретической и профессиональной практической подготовки студентов составляют практические занятия.

Состав и содержание практических работ направлены на реализацию действующего федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.03 Программирование в компьютерных системах.

Ведущей дидактической целью практических занятий является формирование практических умений – профессиональных (умений выполнять определенные действия, операции, необходимые в последующем в профессиональной деятельности), необходимых в последующей учебной деятельности по профессиональным модулям.

В соответствии с рабочей программой ПМ.02. Разработка и администрирование баз данных, МДК.02.01.02 Программное обеспечение компьютерных сетей предусмотрено проведение практических работ.

В результате их выполнения, обучающийся должен:

*уметь:*

- разрабатывать прикладные программы с использованием языка SQL;
- применять стандартные методы для защиты объектов базы данных.

Содержание практических занятий ориентировано на формирование общих компетенций по профессиональному модулю основной профессиональной образовательной программы по специальности:

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.

ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.

ОК 6. Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.

ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.

ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

И овладению профессиональными компетенциями:

ПК.2.1. Разрабатывать объекты базы данных.

ПК.2.2. Реализовывать базу данных в конкретной системе управления базами данных (СУБД).

ПК.2.3. Решать вопросы администрирования базы данных.

ПК.2.4. Реализовывать методы и технологии защиты информации в базах данных.

Выполнение студентами практических работ по ПМ.02. Разработка и администрирование баз данных, МДК.02.01.02 Программное обеспечение компьютерных сетей направлено на:

- обобщение, систематизацию, углубление, закрепление, развитие и детализацию полученных теоретических знаний по конкретным темам междисциплинарных курсов;

- формирование умений применять полученные знания на практике, реализацию единства интеллектуальной и практической деятельности;

- формирование и развитие умений: наблюдать, сравнивать, сопоставлять, анализировать, делать выводы и обобщения;

- развитие интеллектуальных умений у будущих специалистов: аналитических, проектировочных, конструктивных и др.;

- выработку при решении поставленных задач профессионально значимых качеств, таких как самостоятельность, ответственность, точность, творческая инициатива.

Продолжительность выполнения практической работы составляет не менее двух академических часов и проводится после соответствующего занятия, которое обеспечивает наличие знаний, необходимых для ее выполнения.

## 2 МЕТОДИЧЕСКИЕ УКАЗАНИЯ

### Тема 02.01.02 Программное обеспечение компьютерных сетей

#### Практическая работа № 1 Верстка клиентской части сайта.

**Цель работы:** получение практических навыков верстки клиентской части сайта

**Выполнив работу, Вы будете:**

*уметь:*

- верстать сайт;
- разрабатывать прикладные программы с использованием языка SQL;

#### Материальное обеспечение:

Методические указания для выполнения практических работ, персональный компьютер, SublimeText.

#### Задание:

1. Изучить теоретический материал по блочной верстке сайтов. Скопировать предложенные в материале коды, добавив к ним соответствующие теги, согласно приведенным рисункам, посмотреть результат их работы. Стили CSS вынести в отдельный файл (style.css в папку блочная верстка).
2. Изучить работу кодов, предложенных в пример 1 Блочная верстка сайтов. Скопировать предложенные в материале коды и посмотреть результат их работы. Стили CSS вынести в отдельный файл (style.css в папку пример 1).
3. В графическом редакторе разработать логотип группы. Сохранить логотип в папку сайт группы.
4. Сверстать главную страницу, состоящую из 4 блоков: шапка сайта, контейнер, сайдбар, подвал.

#### Краткие теоретические сведения:

**Вёрстка страницы** представляет собой процесс разработки структуры html-документа, результатом которого является веб-страница. Структура веб-страницы определяется соответствующими html-тегами. Теги — прямоугольные блоки-контейнеры для содержимого — не отображаются в окне браузера. Они сообщают браузеру о типе контента, а браузер на основании этой информации выводит на экран их содержимое — текст или медиа-файлы.

Как создать структуру страницы с помощью блоков (блочная вёрстка)

1. Как разбить макет страницы на секции

Чтобы создать макет страницы, необходимо выделить основные разделы (секции) документа.

Стандартная веб-страница содержит следующие секции:

```
<body>
<header></header>
<div class="main"></div>
<footer></footer>
</body>
```

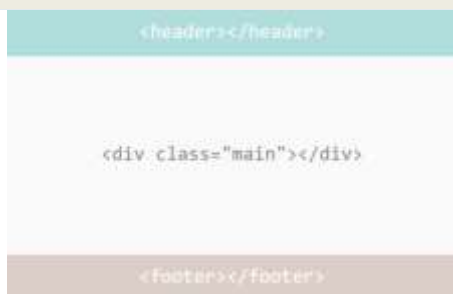


РИС. 1. ОСНОВНЫЕ СЕКЦИИ СТРАНИЦЫ

Элементы <header>, <div> и <footer> — блочные, поэтому они будут занимать всю ширину их блока-контейнера — элемента <body>. Страница с такой разметкой будет хорошо смотреться на небольших экранах, но на устройствах с большим разрешением текст таких страниц будет неудобно читать. Поэтому необходимо добавить тег-контейнер для контента — содержимого каждой секции:

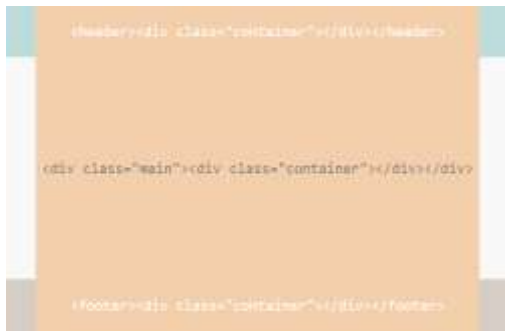


РИС. 2. ОСНОВНЫЕ СЕКЦИИ СТРАНИЦЫ С ТЕГОМ-КОНТЕЙНЕРОМ

и задать для него максимальную ширину, внутренние отступы, которые будут отделять контент от краёв экрана устройств с небольшим разрешением, а также внешние отступы, позволяющие выровнять контейнер по середине родительского блока:

```
.container {
  width: 100%;
  max-width: 1024px; /*максимальная ширина может иметь другое значение*/
  padding: 0 15px;
  margin: 0 auto;
}
```

Таким образом, мы получили каркас для нашей страницы (высота секций в данном случае виртуальная, так как без содержимого секции имеют нулевую высоту).

## 2. Разметка шапки сайта и позиционирование её элементов

Приступим к разметке первой секции, так называемой шапки веб-страницы. Элемент `<header>` предназначен для группировки вводной информации и навигационных средств по странице/сайту. Добавим внутрь тега `<header>` логотип сайта и навигационные ссылки:

```
<header>
  <div class="container">
    <a href="/" class="logo">LOGO</a>
    <nav>
      <ul>
        <li><a href="">Главная</a></li>
        <li><a href="">О нас</a></li>
        <li><a href="">Контакты</a></li>
      </ul>
    </nav>
  </div>
</header>
```

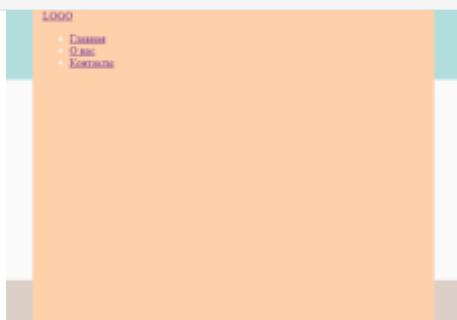


РИС. 3. ШАПКА САЙТА С ДОБАВЛЕННЫМИ ЛОГОТИПОМ И ССЫЛКАМИ

Разместим логотип слева, а ссылки навигации — с выравниванием по правому краю шапки (для наглядности я добавила элементам белую границу):

```
.logo {
  float: left;
}
nav {
```

```

    float: right;
}
nav ul {
    margin: 0;
    padding: 0;
    list-style: none;
}
nav li {
    display: inline-block; /*один из способов разместить элементы в строку*/
}

```

Существует несколько способов разместить блочные элементы в строку.

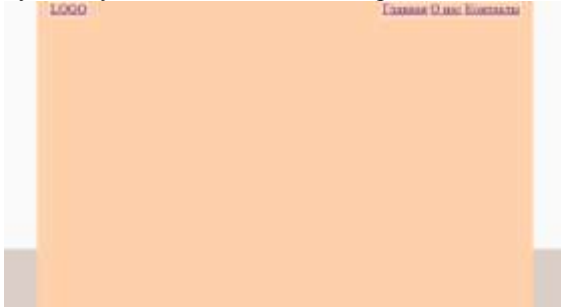


РИС. 4. ЭФФЕКТ СХЛОПЫВАНИЯ БЛОКА-КОНТЕЙНЕРА

Обратите внимание, что после того, как мы применили обтекание, шапка исчезла. Это произошло потому, что плавающие элементы (для которых задано свойство `float`) изымаются из нормального потока и родительский контейнер больше не видит их высоту, поэтому и элемент `<header>` и элемент внутри него с классом `.container` схлопнулись. Чтобы исправить эту ситуацию, воспользуемся очисткой потока для элемента с классом `.container`:

```

.container:after {
    content: "";
    display: table;
    clear: both;
}

```

Также добавим ему вертикальные отступы, отделяющие элементы внутри него от краёв шапки. В результате стили будут иметь следующие вид:

```

.container {
    width: 100%;
    max-width: 1024px;
    padding: 15px;
    margin: 0 auto;
}

```



РИС. 5. ОЧИСТКА ПОТОКА

Рассмотрим ситуацию, когда в качестве логотипа выступает картинка. Она может быть добавлена непосредственно внутрь тега или же в качестве фонового изображения. Картинка будет иметь свою высоту, которая может сильно отличаться от высоты меню навигации, поэтому перед нами встанет проблема вертикального выравнивания элементов шапки.



РИС. 6. ЛОГОТИП-КАРТИНКА

В нашем примере высота логотипа равна 38px, поэтому чтобы выровнять ссылки меню по высоте по середине шапки, нужно задать для них соответствующую высоту строки:

```
nav a {
  text-decoration: none;
  line-height: 38px;
}
```

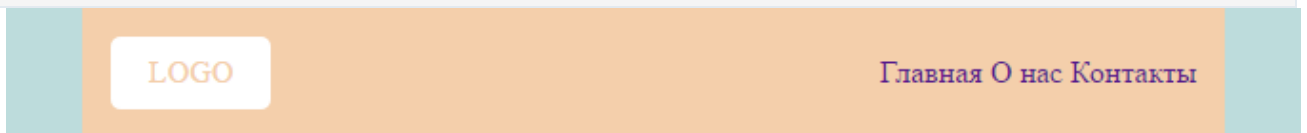


РИС. 7. ВЫРАВНИВАНИЕ ССЫЛОК МЕНЮ ШАПКИ

### 3. Создание сетки для основной части страницы

Основная часть страницы чаще всего представляет собой сетку из блоков разной ширины.

Позиционирование таких блоков осуществляется также с помощью свойства float. Каждую строку блоков обернём дополнительным блоком с классом .row:

```
<div class="main">
  <div class="container">
    <div class="row">
      <div class="col-1-3">1/3</div>
      <div class="col-2-3">2/3</div>
    </div>
    <div class="row">
      <div class="col-1-2">1/2</div>
      <div class="col-1-2">1/2</div>
    </div>
    <div class="row">
      <div class="col-1-4">1/4</div>
      <div class="col-1-4">1/4</div>
      <div class="col-1-2">1/2</div>
    </div>
  </div>
</div>
```



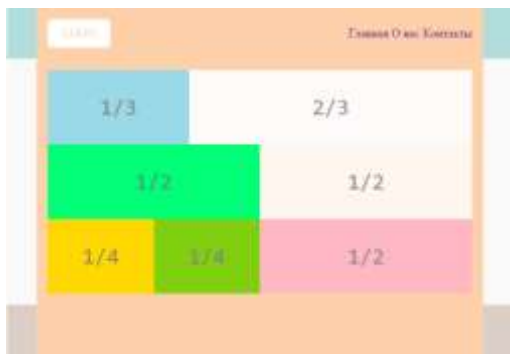


РИС. 7. СЕТКА ОСНОВНОЙ ЧАСТИ СТРАНИЦЫ

```
.col-1-2 {
  width: 50%;
  float: left;
}
.col-1-3 {
  width: 33.3333333333%;
  float: left;
}
.col-1-4 {
  width: 25%;
  float: left;
}
.col-2-3 {
  width: 66.6666666667%;
  float: left;
}
```

Для элемента с классом `.row` также применим очистку потока:

```
.container:after, .row:after {
  content: "";
  display: table;
  clear: both;
}
```

Чтобы отделить ряды друг от друга, можно добавить нижний внешний отступ:

```
.row {
  margin-bottom: 15px;
}
```

Высота блоков сетки определяется высотой их содержимого, поэтому она может быть разной:



РИС. 8. РАЗНАЯ ВЫСОТА ЭЛЕМЕНТОВ СЕТКИ

Высоту блоков можно зафиксировать, указав её явно, например, `.row div {height: 100px}`. Но в этом случае нужно быть уверенным, что при добавлении адаптивности макету содержимое блоков не будет выходить за край блока.

Если необходимо задать фоновый цвет для блоков ряда, то это можно сделать следующим образом: для элемента с классом `.row` добавим новый класс, который позволит стилизовать только этот ряд (получится `.row row-one`), укажем для него фоновый цвет меньшего по высоте блока, а для высокого блока зададим его собственный цвет, то есть таким образом мы сделаем фоновую подложку.

```
<div class="row row-one">
  <div class="col-1-3">Lorem ipsum dolor sit amet.</div>
  <div class="col-2-3">Duis lobortis tempor tortor vel iaculis. Fusce
volutpat commodo lacus, eget vestibulum lorem semper at.</div>
</div>
.row-one {
  background: lightblue;
}
.col-2-3 {
  width: 66.6666666667%;
  float: left;
  background: seashell;
}
```



РИС. 9. ФОНОВАЯ ПОДЛОЖКА

Если основная секция страницы содержит только два блока, то дополнительный ряд-обёртку можно не добавлять:

```
<div class="main">
  <div class="container">
    <div class="col-2-3"></div>
    <div class="col-1-3"></div>
  </div>
</div>
```

#### 4. Разметка подвала страницы

Подвал страницы, или нижний колонтитул, обычно содержит информацию о копирайте, дополнительные ссылки и т.п. Вся эта информация также размещается в столбцах, которые могут быть одинаковой или разной ширины.

```
<footer>
  <div class="container">
    <div class="col-1-3"></div>
    <div class="col-1-3"></div>
    <div class="col-1-3"></div>
  </div>
</footer>
```

Используемый источник:

<https://html5book.ru/vyorstka-stranicy-sayta/>

## Порядок выполнения работы:

### Верстка главной страницы

шапка

ПР-17-1	О группе	Успеваемость	Регистрация	Вход
---------	----------	--------------	-------------	------

Контейнер

Фото группы

Сайдбар

классный руководитель

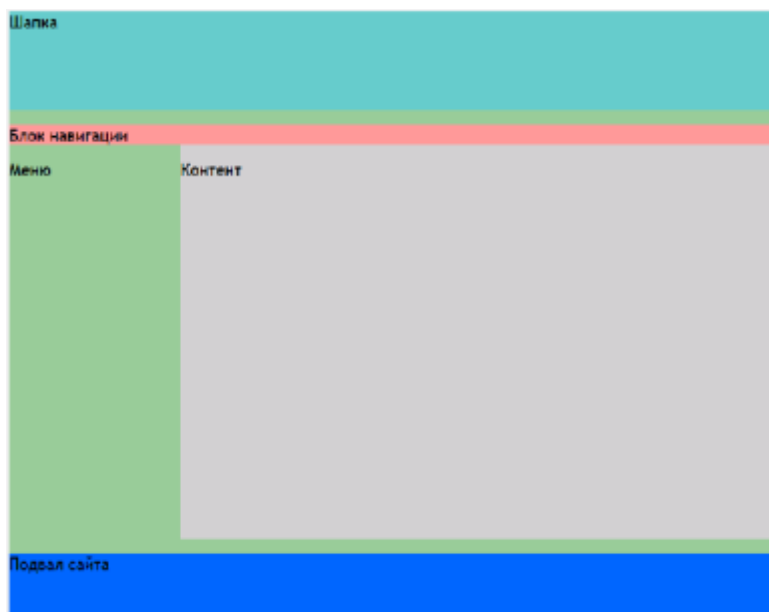
дисциплины  
фотогалерея

Подвал

Разработчик

# Пример1 Блочная верстка сайтов

<p>Файл index.html</p> <pre> &lt;html&gt; &lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=windows 1251" /&gt; &lt;title&gt;Пример блочной верстки&lt;/title&gt; &lt;link rel="stylesheet" type="text/css" href="style.css" /&gt; &lt;/head&gt;  &lt;body&gt;  &lt;div id="container"&gt;   &lt;div id="header"&gt;     &lt;h2&gt;Шапка&lt;/h2&gt;   &lt;/div&gt;    &lt;div id="navigation"&gt;     &lt;h2&gt;Блок навигации&lt;/h2&gt;   &lt;/div&gt;    &lt;div id="menu"&gt;     &lt;h2&gt;Меню&lt;/h2&gt;   &lt;/div&gt;    &lt;div id="content"&gt;     &lt;h2&gt;Контент&lt;/h2&gt;   &lt;/div&gt;    &lt;div id="clear"&gt;    &lt;/div&gt;    &lt;div id="footer"&gt;     &lt;h2&gt;Подвал сайта&lt;/h2&gt;   &lt;/div&gt; &lt;/div&gt;  &lt;/body&gt; &lt;/html&gt; </pre>	<p>Файл style.css</p> <pre> body {   background: #f3f2f3;   color: #000000;   font-family: Trebuchet MS, Arial, Times New Roman;   font-size: 12px; }  #container {   background:#99CC99;   margin: 30px auto;   width: 900px;   height: 600px; }  #header {   background: #66CCCC;   height: 100px;   width: 900px; }  #navigation {   background: #FF9999;   width: 900px;   height: 20px; }  #menu {   background: #99CC99;   float: left;   width: 200px;   height: 400px; }  #content {   background: #d2d0d2;   float: right;    width: 700px;   height: 400px; }  #clear {   clear:both; }  #footer {   background: #0066FF;   height: 80px;   width: 900px; } </pre>
---	--



**Форма представления результата:**  
Отчет по выполненной практической работе.

Практическая работа № 2

Верстка формы регистрации, авторизации на сайте.

**Цель работы:** получение практических навыков верстки формы регистрации, авторизации на сайте

**Выполнив работу, Вы будете:**

*уметь:*

- разрабатывать прикладные программы с использованием языка SQL;
- верстать форму регистрации, авторизации на сайте;

**Материальное обеспечение:**

Методические указания для выполнения практических работ, персональный компьютер, SublimeText.

**Задание:**

Сверстать форму регистрации и авторизации по предложенному ниже образцу.

**Краткие теоретические сведения:**

**Флажок (checkbox)**

`<input type= "checkbox" name= "mycolor" value= "blue">Синий`

**Переключатель (radio)**

`<input type= "radio" name= "mycolor" value= "white">Белый`

**Раскрывающийся список**

`<select name= "Имя списка" size= "Размер" multiple>`

`<option value= "Значение">Первый пункт </option>`

`<option value= "Значение">Второй пункт </option>`

`</select>`

Multiple позволяет выбрать несколько пунктов, удерживая Ctrl.

**Текстовое поле**

`<input type= "text" name= "txtname" size= "10" maxlength= "5" value= "Текст по умолчанию">`

**Текстовая область**

`<form method= "POST">`

`<textarea name= "name" cols= 50 rows= 10 >`

`</textarea><br>`

`<input type= "submit" value= "Отправить">`

`</form>`

**Порядок выполнения работы:**

1.Сверстать форму авторизации

Логин

Пароль

Регистрация

Забыли пароль

Войти

2.Сверстать форму регистрации

Ваше имя:	<input type="text"/>
Ник:	<input type="text"/>
Пароль:	<input type="password"/>
Повтор пароля:	<input type="password"/>
Ваш e-mail	<input type="text"/>
Фотография:	<input type="text"/>
Ваши интересы:	
<input type="checkbox"/> Книги	<input type="checkbox"/> Кухня
<input type="checkbox"/> Музыка	<input type="checkbox"/> Мода
<input type="checkbox"/> Кино	<input type="checkbox"/> Танцы
Ваш пол:	<input type="radio"/> муж <input type="radio"/> жен

Несколько слов о себе

Переслать данные

Сброс данных

### Практическая работа № 3

Разработка формы голосования на сайте с использованием логики в JavaScript.

**Цель работы:** получение практических навыков разработки формы голосования логики с применением JavaScript.

**Выполнив работу, Вы будете:**

*уметь:*

- разрабатывать прикладные программы с использованием языка SQL;
- разрабатывать форму голосования на сайте с использованием логики JavaScript;

**Материальное обеспечение:**

**Задание:**

Разработать форму голосования логики с применением JavaScript.

Методические указания для выполнения практических работ, персональный компьютер, SublimeText.

**Краткие теоретические сведения:**

`<form>...</form>` форма

`<input type= "text">` Текстовое поле

`<input type= "number">` числовое поле

`<input type= "date" value= "2010-12-01">` Дата

`<input type= "checkbox">` флажок

`<input type= "radio">` Радиокнопка

`<input type= "textarea">`

`<input type= "range" min= "0" max= "20">` Создание ползунка

`<input type= "email">` Электронная почта

`<input type= "url">` URL адрес

`<input type= "password">` пароль

Условные операторы в JavaScript.

`=` равно

`!=` не равно

`&&` логическое и

`||` логическое или

`if` (условие)

{

Операторы;

}

`else`

{

Операторы;

}

**Порядок выполнения работы:**

1. Создать на сайте форму голосования.
2. С помощью предложенного ниже примера организовать голосование согласно условию: если поставлена оценка ниже 5, вывести сообщение «Спасибо за Ваше мнение», если оценка больше 5 но меньше 10 – «Благодарим за высокую оценку», оценка больше 10 – «Введена неправильная оценка».

```
<head>
```

```
<script>
```

```
function validate(form)
```

```
{
```

```
if(form.value>=30)
```

```
{
```

```
alert("Да Вы уже взрослый человек!");
```

```
else
```

```
alert("Вы еще повзрослеете");
```

```
}
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<form name= "Form_1">  
Ваш возраст:<input type= "text" size= 5 name= "age">  
<input type= "button" value= "Подтвердите" onClick= "validate (this.form.age)">  
</form>  
</body>
```

**Форма представления результата:**

Отчет по выполненной практической работе.



## Тема 02.01.02 Программное обеспечение компьютерных сетей

### Практическая работа № 4

Внедрение canvas в форму голосования.

**Цель работы:** получение практических навыков работы с canvas

**Выполнив работу, Вы будете:**

*уметь:*

- разрабатывать прикладные программы с использованием языка SQL;
- внедрять canvas в форму голосования с использованием JavaScript.

**Материальное обеспечение:**

Методические указания для выполнения практических работ, персональный компьютер, SublimeText.

**Задание:**

С помощью canvas внедрить смайл в форму голосования.

**Краткие теоретические сведения:**

Canvas (англ. canvas — «холст», рус. канвас) — элемент HTML5, предназначенный для создания растрового двухмерного изображения при помощи скриптов, обычно на языке JavaScript. Начало отсчёта блока находится слева сверху. От него и строится каждый элемент блока. Размер пространства координат не обязательно отражает размер фактической отображаемой площади. По умолчанию его ширина равна трёмстам пикселям, а высота ста пятидесяти.

Используется, как правило, для отрисовки графиков для статей и игрового поля в некоторых браузерных играх. Но также может использоваться для встраивания видео в страницу и создания полноценного плеера.

**Порядок выполнения работы:**

С помощью приведенного ниже скрипта, организовать внедрение смайлика в форму для голосования на сайте. Причем, если поставлена оценка ниже 5, вывести грустный смайлик, если оценка больше 5 но меньше 10, вывести хмурый смайлик, оценка больше 10 – веселый смайлик.

```
<!Doctype html>
<html>
<head>
<title>canvas</title>
<meta charset= "UTF-8">
<script src= "smile.js">
</script>
</head>
<body>
<form name= "form">
<input type= "text" name= "name">
<input type= "button" onClick= "validate (this.value)" value= "Проверить">
</form>
<canvas id= "smile" width= "200" height= "200">
<p>Ваш браузер</p>
</canvas>
</body>
</html>
```

```
smile.js
function validate()
{
var a=document.forms["form"].name.value;
if (a>5)
{
var drawingCanvas=document.getElementById('smile');
if (drawingCanvas && drawingCanvas.getContext)
{
var context=drawingCanvas.getContext('2d');
context.strokeStyle="#000";
context.fillStyle="#fc0";
context.beginPath();
```

```

context.arc(100,100,50,0,Math.PI*2,true);
context.closePath();
context.stroke();
context.fill();

context.fillStyle="#fff";
context.beginPath();
context.arc(84,90,8,0,Math.PI*2,true);
context.closePath();
context.stroke();
context.fill();

context.beginPath();
context.arc(116,90,8,0,Math.PI*2,true);
context.closePath();
context.stroke();
context.fill();

context.beginPath();
context.moveTo(70,115);
context.quadraticCurveTo(100,130,130,115);
context.quadraticCurveTo(100,150,70,115);
context.closePath();
context.stroke();
context.fill();
}
}

```

If ....продолжить логику самостоятельно.

#### **Форма представления результата:**

Отчет по выполненной практической работе.

## Тема 02.01.02 Программное обеспечение компьютерных сетей

### Практическая работа № 5

Добавление таймера на сайт с использованием возможностей JavaScript.

**Цель работы:** получение практических навыков добавления таймера на сайт.

**Выполнив работу, Вы будете:**

*уметь:*

- разрабатывать прикладные программы с использованием языка SQL;
- внедрять таймер на сайт с использованием JavaScript.

**Материальное обеспечение:**

Методические указания для выполнения практических работ, персональный компьютер, SublimeText.

**Краткие теоретические сведения:**

```
<head>
<script>
function JSClock(){
var time = new Date()
var hour = time.getHours()
var minute = time.getMinutes()
var second = time.getSeconds()
var temp = "" + hour
temp +=((minute<10)?"0":"")+minute
temp +=((second<10)?"0":"")+second
document.clockForm.digits.value=temp
id=setTimeout("JSClock()",1000)
}
</script>
</head>
<body bgcolor= "#FFCCOO" onLoad= "JSClock()">
<form name= "clockForm">
<input type= "text" name= "digits" size= 8 value= "">
</form>
</body>
```

**Задание:**

С помощью приведенного выше скрипта добавить часы в подвал сайта.

**Форма представления результата:**

Отчет по выполненной практической работе.

## Тема 02.01.02 Программное обеспечение компьютерных сетей

### Практическая работа № 6

Добавление анимации на сайт. Перемещение логотипа.

**Цель работы:** получение практических навыков работы с анимацией.

**Выполнив работу, Вы будете:**

*уметь:*

- разрабатывать прикладные программы с использованием языка SQL;
- внедрять анимацию на сайт.

#### Материальное обеспечение:

Методические указания для выполнения практических работ, персональный компьютер, SublimeText.

**Задание:** Добавить анимацию на сайт в виде перемещения логотипа

#### Краткие теоретические сведения:

Пример №1 Перемещение рисунка на экране (паровозика)

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="style.css">
</head>
<body>
  
</body>
</html>
```

Файл style.css

```
.train {
  position: relative;
  cursor: pointer;
  width: 177px;
  height: 160px;
  left: 100px;
  transition: left 5s cubic-bezier(.5, -1, .5, 2);
}
```

Пример №2

Осуществить передвижение паровозика на расстояние 1000 px

Пример №3

Осуществить абсолютное позиционирование паровозика, задав отступ сверху 300 px, справа 200 px.

Пример №4

Ускорить передвижение паровозика

Пример №5

Увеличить паровозик, сделав его размер по горизонтали и вертикали 500 px.

Пример №6

Осуществить передвижение своей картинке, задав свои координаты.

Пример №7 Перемещение кнопки по горизонтали

```
<html>
<div class="progress"></div>
<style>
  /* Современные браузеры, кроме Chrome, Opera, Safari */
  @keyframes go-left-right { /* назовём анимацию: "go-left-right" */
    from {
      left: 0px; /* от: left: 0px */
    }
    to {
      left: calc(100% - 50px); /* до: left: 100%-50px */
    }
  }

  /* Префикс для Chrome, Opera, Safari */
  @-webkit-keyframes go-left-right {
    from {
```

```

    left: 0px;
  }
  to {
    left: calc(100% - 50px);
  }
}

.progress {
  /* применить анимацию go-left-right */
  /* продолжительность 3s */
  /* количество раз: бесконечное (infinite) */
  /* менять направление анимации каждый раз (alternate) */
  animation: go-left-right 3s infinite alternate;
  -webkit-animation: go-left-right 3s infinite alternate;

  position: relative;
  border: 2px solid green;
  width: 50px;
  height: 20px;
  background: lime;
}
</style>
</html>

```

**Порядок выполнения работы:**

1. Проверить работу предложенных скриптов
2. С помощью предложенного ниже скрипта добавить анимацию на сайт (осуществить перемещение логотипа сайта).

**Форма представления результата:**

Отчет по выполненной практической работе.

## Тема 02.01.02 Программное обеспечение компьютерных сетей

### Практическая работа № 7

Добавление информации на сайт с использованием массивов.

**Цель работы:** получение практических навыков работы с массивами.

**Выполнив работу, Вы будете:**

*уметь:*

- разрабатывать прикладные программы с использованием языка SQL;
- добавлять на сайт информацию из массивов.

**Материальное обеспечение:**

Методические указания для выполнения практических работ, персональный компьютер, SublimeText.

**Задание:**

С помощью массива организовать вывод информации на сайт:

Вкладка «Успеваемость»

Фамилия	Предмет	Оценка

Вкладка «Дисциплины»

Фамилия преподавателя	Предмет	Кабинет

Вкладка «Фотогалерея»

Организовать вывод 9 фото.

## Тема 02.01.02 Программное обеспечение компьютерных сетей

### Практическая работа № 8

Разработка обработчиков событий JavaScript на сайте.

**Цель работы:** получение практических навыков работы с событиями JavaScript.

**Выполнив работу, Вы будете:**

*уметь:*

- разрабатывать прикладные программы с использованием языка SQL;
- увеличивать/уменьшать фотографии на сайте.

#### **Материальное обеспечение:**

Методические указания для выполнения практических работ, персональный компьютер, SublimeText.

#### **Краткие теоретические сведения:**

Событию можно назначить обработчик, то есть функцию, которая срабатывает, как только событие произошло. Именно благодаря обработчикам JavaScript-код может реагировать на действия посетителя.

Пример

```
<!Doctype html>
<html>
<head>
<script>
var bigsize= "300";// Размер большой картинки
var smallsize= "150";// Размер маленькой картинки
function changeSizeImage(im)
{
if (im.height==bigsize)
im.height=smallsize;
else im.height=bigsize;
}
</script>
</head>
<img src= "4.png" height= "150" onClick= "changeSizeImage (this)">
</body>
</html>
```

#### **Задание:**

Выполнить увеличение картинок в фотогаллере на сайте по наведению мыши или по щелчку.

#### **Форма представления результата:**

Отчет по выполненной практической работе.

## Тема 02.01.02 Программное обеспечение компьютерных сетей

### Практическая работа № 9

Разработка базы данных средствами PHP Myadmin.

**Цель работы:** получение практических навыков разработки базы данных средствами PHP Myadmin.

**Выполнив работу, Вы будете:**

*уметь:*

- разрабатывать прикладные программы с использованием языка SQL;
- разрабатывать базу данных средствами PHP Myadmin.

**- Материальное обеспечение:**

Методические указания для выполнения практических работ, персональный компьютер, SublimeText, Open Server

**Краткие теоретические сведения:**

1. Зайти в приложение PHP Myadmin.

а) запустить сервер

б) в адресной строке браузера: localhost/OpenServer/PhpMyadmin (или пкм на флажке справа на панели задач, выбрать дополнительно, PhpMyadmin.

2. Создать новую базу данных, задав ей имя avto.

3. Создать таблицу postavshiki, в строке Поля указать 4, нажать Пошел.

4. Создать поля: id, name\_postavshika, adres, nationalnost. Вид полей: для id int, для остальных varchar, длина 100. Поле id является первичным ключом. Нажать кнопку сохранить.

5. Нажать вкладку вставить, заполнить поля данными. Для набора очередной записи активировать вкладку Вставить новую запись. Создать 5 записей.

6. Щелкнуть имя базы данных avto. Создать таблицу mashini с полями: id, name\_postavshika, id\_mashini, marka\_mashini, cena\_mashini. Создать 7 записей.

7. Создать таблицу prodagi с полями: id\_mashini, name\_pokupatelya, marka\_mashini. Создать 7 записей.

8. Активировать базу данных avto, зайти во вкладку Еще -> Дизайнер -> Показать таблицы -> Связи между таблиц.

8. Активировать таблицу mashini. Создать к ней SQL запросы:

1) из таблицы mashini выбираются машины с ценой более 500000

SELECT\*FROM 'mashini' WHERE cena\_mashini>500000

2) из таблицы mashini выбираются машины с ценой между 400000 и 500000

SELECT\*FROM 'mashini' WHERE cena\_mashini BETWEEN 400000 AND 500000

3) из таблицы mashini выбираются машины с маркой Shkoda

SELECT\*FROM 'mashini' WHERE marka\_mashini= 'Shkoda'

4) из таблицы postavshiki выбираются поставщики, названия которых заканчиваются на в

SELECT\*FROM 'postavshiki' WHERE name\_postavshika LIKE '%в'

**Задание:**

Согласно предметной области «Учебная группа» спроектировать и разработать средствами PHP Myadmin базу данных.

**Форма представления результата:**

Отчет по выполненной практической работе.



## Тема 02.01.02 Программное обеспечение компьютерных сетей

### Практическая работа № 10

Добавление информации на сайт из базы данных средствами PHP.

**Цель работы:** получение практических навыков добавление из базы данных средствами PHP информации на сайт.

**Выполнив работу, Вы будете:**

*уметь:*

- разрабатывать прикладные программы с использованием языка SQL;
- добавлять из базы данных средствами PHP информацию на сайт.

**- Материальное обеспечение:**

Методические указания для выполнения практических работ, персональный компьютер, SublimeText, Open Server.

**Краткие теоретические сведения:**

```
<!DOCTYPE html>
<html>
<head>
<meta charset= "utf-8">
</head>
<?php
//Проверка правильности подключения
try
{
Подключение к базе данных
$pdo=new PDO('mysqlhost=localhost;dbname=avto', 'root', '');
}
catch (PDOException $e)
{
die ('Подключение не удалось'. $e->getMessage());
}
//Формируем запрос
$sql= "SELECT*FROM postavshiki";
$cat=$pdo->query($sql);
//Перебор и вывод результатов
while ($row=$cat->fetch())
{
echo $row['name'];
}
?>
</html>
```

**Задание:**

Используя предложенный выше скрипт осуществить вывод на сайт из базы данных:

Вкладка «Успеваемость»

Фамилия	Предмет	Оценка

Вкладка «Дисциплины»

Фамилия преподавателя	Предмет	Кабинет

Вкладка «Фотогалерея»

Организовать вывод 9 фото.

**Форма представления результата:**

Отчет по выполненной практической работе.

## Тема 02.01.02 Программное обеспечение компьютерных сетей

### Практическая работа № 11

Подключение формы регистрации на сайте средствами PHP.

**Цель работы:** получение практических навыков подключения формы регистрации средствами PHP.

**Выполнив работу, Вы будете:**

*уметь:*

- разрабатывать прикладные программы с использованием языка SQL;
- подключать форму регистрации средствами PHP.

**- Материальное обеспечение:**

Методические указания для выполнения практических работ, персональный компьютер, SublimeText, Open Server.

**Краткие теоретические сведения:**

```
<!DOCTYPE html>
<html>
<head>
<meta charset= "utf-8">
</head>
<body>
<form method= "POST">
Фамилия:<input type= "text" name= "fami"><br>
Имя:<input type= "text" name= "imy"><br>
<input type= "submit" value= "ок">
</form>
<?php
$a=$_POST['fami'];
$b=$_POST['imy'];
try
{
Подключение к базе данных
$pdo=new PDO('mysqlhost=localhost;dbname=tehnika', 'root', '');
}
catch (PDOException $e)
{
die ('Подключение не удалось'. $e->getMessage());
}
//Формируем запрос
$sql= "INSERT INTO sotrudniki (fam,name) values('$a', '$b')";
$cat=$pdo->query($sql);
?>
```

**Задание:**

Подключить форму регистрации к базе данных

**Форма представления результата:**

Отчет по выполненной практической работе.

Ваше имя:	
Ник:	
Пароль:	
Повтор пароля:	
Ваш e-mail	
Фотография:	
Ваши интересы:	
<input type="checkbox"/> Книги	<input type="checkbox"/> Кухня
<input type="checkbox"/> Музыка	<input type="checkbox"/> Moda
<input type="checkbox"/> Кино	<input type="checkbox"/> Танцы
Ваш пол:	<input type="radio"/> Муж <input type="radio"/> Жен

Несколько слов о себе

Перестать данные

Сброс данных

## Тема 02.01.02 Программное обеспечение компьютерных сетей

### Практическая работа № 12

Подключение формы авторизации на сайте средствами PHP.

**Цель работы:** получение практических навыков подключения формы авторизации средствами PHP.

**Выполнив работу, Вы будете:**

*уметь:*

- разрабатывать прикладные программы с использованием языка SQL;
- подключать форму авторизации средствами PHP.

**- Материальное обеспечение:**

Методические указания для выполнения практических работ, персональный компьютер, SublimeText, Open Server.

**Краткие теоретические сведения:**

```
<!DOCTYPE html>
<html>
<head>
<meta charset= "utf-8">
</head>
<body>
<form method= "POST">
Фамилия:<input type= "text" name= "fami"><br>
Имя:<input type= "text" name= "imy"><br>
<input type= "submit" value= "ок">
</form>
<?php
$a=$_POST['fami'];
$b=$_POST['imy'];
try
{
Подключение к базе данных
$pdo=new PDO('mysqlhost=localhost;dbname=tehnika', 'root', '');
}
catch (PDOException $e)
{
die ('Подключение не удалось'. $e->getMessage());
}
//Формируем запрос
$sql= "SELECT*FROM sotrudniki";
$cat=$pdo->query($sql);
//Перебор и вывод результатов
while ($row=$cat->fetch())
{
if ($row['fam']==$a && $row['name']==$b)
{
$c=1;
}
}
if ($c==1)
{
echo "наш пользователь";
}
else
{
echo "такого пользователя нет";
}
?>
```

**Задание:**

Подключить форму авторизации к базе данных

Логин

Пароль

[Регистрация](#)

**Форма представления результата:**  
Отчет по выполненной практической работе.

## Тема 02.01.02 Программное обеспечение компьютерных сетей

### Практическая работа № 13

Разработка формы удаления, обновления информации средствами PHP.

**Цель работы:** получение практических навыков удаления, обновления информации средствами PHP.

**Выполнив работу, Вы будете:**

*уметь:*

- разрабатывать прикладные программы с использованием языка SQL;
- удалять, обновлять информацию средствами PHP.

**- Материальное обеспечение:**

Методические указания для выполнения практических работ, персональный компьютер, SublimeText, Open Server.

**Краткие теоретические сведения:**

Удаление данных из таблиц:

Исходные данные:

```
DELETE FROM lessons  
WHERE lessons.date ='2016-10-02'
```

Оператор Update

Запрос предназначен для обновления (изменения) существующих данных. Update в переводе с английского языка — обновить.

```
UPDATE teachers
```

```
SET
```

```
zarplata = zarplata * 2,
```

```
premia = premia * 10
```

**Задание:**

Напротив каждой строки таблицы из вкладки «Успеваемость», «Дисциплины» и «Фотогалерея» создать кнопки для удаления и обновления

информации в базе. С помощью PHP и SQL запрограммировать эти кнопки.

Вкладка «Успеваемость»

Фамилия	Предмет	Оценка		
Иванов	Физкультура	5	<input type="button" value="удалить"/>	<input type="button" value="изменить"/>
Петров	История	4	<input type="button" value="удалить"/>	<input type="button" value="изменить"/>

**Форма представления результата:**

Отчет по выполненной практической работе.

## Тема 02.01.02 Программное обеспечение компьютерных сетей

### Практическая работа № 14

Разграничение прав доступа на сайте.

**Цель работы:** получение практических навыков разграничения прав доступа на сайте.

**Выполнив работу, Вы будете:**

*уметь:*

- разрабатывать прикладные программы с использованием языка SQL;
- разграничивать доступ на сайте.

**- Материальное обеспечение:**

Методические указания для выполнения практических работ, персональный компьютер, SublimeText, Open Server.

**Краткие теоретические сведения:**

Очень часто на сайтах делают различные типы пользователей.

Каждый тип обладает разными привилегиями.

Самый простой вариант, который можно реализовать - это два типа пользователей: обычный юзер и администратор.

Более сложные варианты: юзер, редактор, публикатор, модератор, дизайнер, администратор.

Конечно, количество типов пользователей зависит конкретно от вашего сайта.

Обычно права доступа хранят в базе данных в специальном поле.

Давайте назовем его status.

Сделаем сайт, на котором только два типа пользователей: юзер и администратор.

Пусть юзеру соответствует статус '1' и самый низкий приоритет, а администратору - статус '10' и самый высокий приоритет.

Админка

Админкой называется зона сайта, куда имеет доступ только администратор сайта (иногда, некоторые другие типы пользователей могут иметь ограниченный доступ).

В админке обычно реализуется добавление и редактирование текстов статей, управление пользователями (удаление, бан, изменение статуса) и многое другое.

Как ограничить доступ в админку только для администратора, вы уже знаете – проверяем авторизованность и проверяем то, что пользователь имеет статус администратора.

Если оба условия выполняются - даем доступ в админку.

Иногда вход в админку защищают дополнительным логином-паролем. И это правильно – админка самая важная часть сайта с точки зрения хакера!

У нас в таблице users не зря было два пользователя с характерными usernames user и admin – назначим им теперь соответствующие статусы:

id	login (Логин)	password (Соленый пароль)	cookie	status (Статус)
	user	827ccb0eea8a706c4c34a16891f84e7b	sdfLjgyl	1
	admin	01cfcd4f6b8770febf40cb906715822	sMtrnwpJ	10

Теперь при авторизации мы должны добавить значение поля status в переменную сессии \$\_SESSION['status']:

```
<?php
```

```
...
```

```
//Стартуем сессию:  
session_start();
```

```
//Пишем в сессию информацию о том, что мы авторизовались:  
$_SESSION['auth'] = true;
```

```
/*
```

```
    Пишем в сессию login и id пользователя  
    (их мы берем из переменной $user!):
```

```
*/
```

```
$_SESSION['id'] = $user['id'];  
$_SESSION['login'] = $user['login'];
```

```
//Пишем в сессию статус пользователя (приоритет):  
$_SESSION['status'] = $user['status'];
```

...

?>

Теперь, если какое-то действие требует определенного приоритета, мы можем проверить его так:

```
<?php
```

```
//Если пользователь авторизован и администратор, то выполним какой-то код:
```

```
if ( $_SESSION['auth'] == true and $_SESSION['staus'] == 10) {
```

```
}
```

?>

При регистрации нового пользователя мы должны жестко задать его статус (скорее всего это будет 1) при сохранении в базу данных (чтобы хакер не смог зарегистрироваться как админ!).

```
<?php
```

```
/*
```

При регистрации устанавливаем статус в единицу:

```
*/
```

```
$query = 'INSERT INTO users SET login="'.$login.'", password="'.$saltedPassword.'", salt="'.$salt.'", status=1';
```

```
?>
```

### **Задание:**

Осуществить на сайте разграничение прав доступа. Создать две роли: пользователь (дать ему определенные возможности) и администратор (у него свои возможности).

### **Форма представления результата:**

Отчет по выполненной практической работе.

## Тема 02.01.02 Программное обеспечение компьютерных сетей

### Практическая работа № 15

Cookie, сессии на сайте.

**Цель работы:** получение практических навыков разграничения прав доступа на сайте.

**Выполнив работу, Вы будете:**

*уметь:*

- разрабатывать прикладные программы с использованием языка SQL;
- разграничивать доступ на сайте.

**- Материальное обеспечение:**

Методические указания для выполнения практических работ, персональный компьютер, SublimeText, Open Server.

**Краткие теоретические сведения**

Cookie

PHP прозрачно поддерживает HTTP cookies. Cookies - это механизм хранения данных браузером удаленной машины для отслеживания или идентификации возвращающихся посетителей. Вы можете установить cookies при помощи функций `setcookie()` или `setrawcookie()`. Cookies являются частью HTTP-заголовка, поэтому `setcookie()` должна вызываться до любого вывода данных в браузер. Это то же самое ограничение, которое имеет функция `header()`. Вы можете использовать функции буферизации вывода, чтобы задержать вывод результатов работы скрипта до того момента, когда будет известно, понадобится ли установка cookies или других заголовков.

Любые cookies, отправленные серверу браузером клиента, будут автоматически включены в суперглобальный массив `$_COOKIE`, если директива `variables_order` содержит букву "C". Для назначения нескольких значений одной cookie, просто добавьте [] к её имени.

В зависимости от значения опции `register_globals`, cookies могут быть представлены обычными PHP-переменными. Однако рекомендуется не полагаться на эту возможность, так как она обычно отключена в целях безопасности.

Сессии

Сессии являются простым способом хранения информации для отдельных пользователей с уникальным идентификатором сессии. Это может использоваться для сохранения состояния между запросами страниц. Идентификаторы сессий обычно отправляются браузеру через сессионный cookie и используются для получения имеющихся данных сессии. Отсутствие идентификатора сессии или сессионного cookie сообщает PHP о том, что необходимо создать новую сессию и сгенерировать новый идентификатор сессии.

Сессии используют простую технологию. Когда сессия создана, PHP будет либо получать существующую сессию, используя переданный идентификатор (обычно из сессионного cookie) или, если ничего не передавалось, будет создана новая сессия. PHP заполнит суперглобальную переменную `$_SESSION` сессионной информацией после того, как будет запущена сессия. Когда PHP завершает работу, он автоматически сериализует содержимое суперглобальной переменной `$_SESSION` и отправит для сохранения, используя сессионный обработчик для записи сессии.

По умолчанию PHP использует внутренний обработчик `files` для сохранения сессий, который установлен в INI-переменной `session.save_handler`. Этот обработчик сохраняет данные на сервере в директории, указанной в конфигурационной директиве `session.save_path`.

Сессии могут запускаться вручную с помощью функции `session_start()`. Если директива `session.auto_start` установлена в 1, сессия автоматически запустится, в начале запроса.

Сессия обычно завершает свою работу, когда PHP заканчивает исполнять скрипт, но может быть завершена и вручную с помощью функции `session_write_close()`.

Пример #1 Регистрация переменной с помощью `$_SESSION`.

---

```
<?php
session_start();
if (!isset($_SESSION['count'])) {
    $_SESSION['count'] = 0;
} else {
    $_SESSION['count']++;
}
?>
```

---

Пример #2 Отмена объявления переменной с помощью `$_SESSION`.



---

```
<?php
session_start();
unset($_SESSION['count']);
?>
```

**Задание:**

Организовать на сайте структуру:

Вход по логину и паролю -> Доступ к сайту -> По кнопке

Выход

осуществить отмену объявления переменной с помощью \$\_SESSION

**Форма представления результата:**

Отчет по выполненной практической работе.