



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ИНСТИТУТ ТЕХНОЛОГИЙ (ФИЛИАЛ) ФЕДЕРАЛЬНОГО
ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»
В Г. ВОЛГОДОНСКЕ РОСТОВСКОЙ ОБЛАСТИ**

(Институт технологий (филиал) ДГТУ в г. Волгодонске)

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

для проведения лабораторного практикума

по дисциплине

«Методы и средства проектирования информационных систем и технологий»

для обучающихся по направлению подготовки

09.03.02 Информационные системы и технологии

программа бакалавриата «Информационные системы»

г. Волгодонск

2021

ОГЛАВЛЕНИЕ

Практическая работа № 1. Создание форм. Основы HTML.....	3
1.1 Задания.....	3
1.2 Теоритический материал.....	3
1.3 Пример создания формы.....	8
Практическая работа № 2. Создание баз данных с помощью MySQL и phpMyAdmin.....	13
2.1 Задания.....	13
2.2 Теоритический материал.....	13
Практическая работа № 3. Формирование запросов к базе данных SQL.....	19
3.1 Задания.....	19
3.2 Теоритический материал.....	19
Практическая работа № 4. Изучение среды разработки Web приложений.....	24
4.1 Задания.....	24
4.2 Теоритический материал.....	24
Практическая работа № 5. Каскадные таблицы стилей (CSS).....	34
5.1 Задания.....	34
5.2 Теоретический материал.....	34
Практическая работа № 6. Передача параметров от HTML формы к PHP процессу.....	51
6.1 Задания.....	51
6.2 Теоретический материал.....	51
Практическая работа № 7. Программное формирование запросов к базе данных SQL из программы на языке PHP.....	54
7.1 Задания.....	54
7.2 Теоритический материал.....	54
ЛИТЕРАТУРА.....	58

Практическая работа № 1. Создание форм. Основы HTML.

Цель: изучить создание форм с помощью HTML и ввод данных.

1.1 Задания.

Для заданной вариантном предметной области построить интерфейсную форму. Вариант определяется в приведенной ниже таблице по последнему номеру зачетной книжки студента.

Последняя цифра в зачетной книжке студента	Номер варианта	Предметная область
1	1	Мониторы
2	2	Принтеры
3	3	Планшеты
4	4	Ноутбуки
5	5	Манипуляторы «мышь»
6	6	Центральные процессорные блоки
7	7	Смартфоны
8	8	Коммутаторы (хабы)
9	9	Клавиатуры
0	10	Сканеры

1.2 Теоритический материал.

Форма — это инструмент, с помощью которого HTML-документ может послать некоторую информацию в некоторую заранее определенную точку внешнего мира, где информация будет некоторым образом обработана. Форма — это инструмент, с помощью которого HTML-документ может послать некоторую информацию в некоторую заранее определенную точку внешнего мира, где информация будет некоторым образом обработана.

1) Создание простой формы

Теги `<form>` и `</form>` задают начало и конец формы. Начинаящий форму тег `<form>` содержит два атрибута: *action* и *method*. Атрибут *action* содержит адрес URL сценария, который должен быть вызван для обработки сценария. Атрибут *method* указывает браузеру, какой вид HTTP запроса

необходимо использовать для отправки формы; возможны значения *POST* и *GET*.

Замечание

Главное отличие методов *POST* и *GET* заключается в способе передачи информации. В методе *GET* параметры передаются через адресную строку, т.е. по сути в HTTP-заголовке запроса, в то время как в методе *POST* параметры передаются через тело HTTP-запроса и никак не отражаются на виде адресной строки.

```
<form method="post" action="../admin/add_story.php">
</form>
```

2) Флажок (*checkbox*)

Флажки *checkbox* предлагают пользователю ряд вариантов, и разрешает выбор нескольких из них.

```
<input name="Имя переключателя" type="Тип" value="Значение">
```

Группа флажков состоит из элементов *<input>*, имеющих одинаковые атрибуты *name* и *type(checkbox)*. Если мы хотим, чтобы элемент был отмечен по умолчанию необходимо пометить его как *checked*. Если элемент выбран, то сценарию поступит строка *имя=значение*, в противном случае в обработчик формы не *придет ничего*, т.е. не выбранные флажки вообще никак не проявляют себя в переданном наборе данных.

Пример:

```
<input name="mycolor" type="checkbox" value="red" checked>Красный(
выбран по умолчанию)
<input name="mycolor" type="checkbox" value="blue">Синий
<input name="mycolor" type="checkbox" value="black">Черный
<input name="mycolor" type="checkbox" value="white">Белый
```

3) Переключатель(*radio*)

Переключатели *radio* предлагают пользователю ряд вариантов, но разрешает выбрать только один из них.

```
<input name="Имя переключателя" type="Тип" value="Значение">
```

Переключатель (*radio*) имеет атрибуты *name*, *type* и *value*. Атрибут *name* задает имя переключателя, *type* задает тип *radio*, а атрибут *value* задает значение. Если пользователь выберет переключатель, то

сценарию будет передана строка *имя=значение*. При необходимости можно указать параметр *checked*, который указывает на то, что переключатель будет иметь фокус (т.е. будет отмечен по умолчанию) при загрузке страницы. Переключатели также можно объединять в группы, для этого они должны иметь одно и то же имя.

Пример:

```
<input name="mycolor" type="radio" value="white"> Белый  
<input name="mycolor " type="radio" value="green" checked> Зеленый  
(выбран по умолчанию)  
<input name="mycolor " type="radio" value="blue"> Синий  
<input name="mycolor " type="radio" value="red"> Красный  
<input name="mycolor " type="radio" value="black"> Черный
```

4) Кнопка сброса формы(Reset)

```
<input type="Тип" name="Имя кнопки" value="Надпись на кнопке">
```

При нажатии на кнопку сброса(*reset*), все элементы формы будут установлены в то состояние, которое было задано в атрибутах по умолчанию, причем отправка формы *не производится*.

Пример:

```
<input type="reset" name="Reset" value="Очистить форму">
```

5) Выпадающий список (select)

Тэг *<select>* представляет собой выпадающий или раскрытый список, при этом одновременно могут быть выбраны одна или несколько строк.

Список начинается с парных тегов *<select></select>*. Теги *<option></option>* позволяют определить содержимое списка, а параметр *value* определяет значение строки. Если в теге *<option>* указан параметр *selected*, то строка будет изначально выбранной. Параметр *size* задает, сколько строк будет занимать список. Если *size* равен 1, то список будет выпадающим. Если указан атрибут *multiple*, то разрешено выбирать несколько элементов из списка(при *size = 1* не имеет смысла).

```
<select name="Имя списка" size = "Размер" multiple>  
<option value="Значение">Отображаемый текст в списке</option>  
</select>
```

При передаче данных выпадающего списка сценарию передается строка *имя=значение*, а при раскрытом списке передается строка *имя=значение1&имя=значение2&имя=значениеN*.

6) Текстовое поле (text)

Позволяет пользователям вводить различную информацию.

```
<input type="Тип" name="Имя поля" size="Размер" maxlength="Макс. количество символов">
```

При создании обычного текстового поля размером *size* и максимальной допустимой длины *maxlength* символов, атрибут *type* принимает значение *text*. Если указан параметр *value*, то поле будет содержать отображать *value*-текст. При создании поля не забывайте указывать имя поля, т.к. этот атрибут является обязательным.

Пример:

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

7) Поле для ввода пароля (password)

Полностью аналогичен текстовому полю, за исключением того что символы, набираемые пользователем, не будут отображаться на экране.

Пример:

```
<input type="password" name="txtName" size="10" maxlength="5">
```

8) Многострочное поле ввода текста (textarea)

Многострочное поле ввода текста позволяет отправлять не одну строку, а сразу несколько. По умолчанию тег создает пустое поле шириной в 20 символов и состоящее из двух строк.

```
<textarea name="Имя поля" cols="Ширина поля" rows="Число строк">Текст</textarea>
```

Многострочное поле ввода текста начинается с парных тегов `<textarea>``</textarea>`. Тэг `name` задает имя многострочного поля. Также можно указать ширину поля (`cols`) и число строк (`rows`). При необходимости можно указать атрибут `readonly`, который запрещает редактировать, удалять и изменять текст, т.е. текст будет предназначен только для чтения. Если необходимо чтобы текст был изначально отображен

в многострочном поле ввода, то его необходимо поместить между тэгами `<textarea></textarea>`.

Пример:

`<textarea name="txtArea" cols="15" rows="10" readonly>` Текст, который изначально будет отображен в многострочном поле ввода и который нельзя изменять, т.к. указан атрибут `readonly` `</textarea>`

9) Скрытое текстовое поле

Позволяет передавать сценарию какую то служебную информацию, не отображая её на странице.

`<input name="Имя" type="Тип" value="Значение">`

Скрытое поле начинается с тега `<input>`, атрибуты которого являются *name*, *type* и *value*. Атрибут *name* задает имя поля, *type* определяет тип поля, а атрибут *value* задает значение поля.

Пример:

`<input name="email" type="hidden" value="spam@nosпам.ru">`

10) Кнопка отправки формы (submit)

Служит для отправки формы сценарию.

`<input type="Тип" name="Имя кнопки" value="Текст кнопки">`

При создании кнопки для отправки формы необходимо указать 2 атрибута: *type="submit"* и *value="Текст кнопки"*. Атрибут *name* необходим если кнопка не одна, а несколько и все они созданы для разных операций, например кнопки "Сохранить", "Удалить", "Редактировать" и т.д. После нажатия на кнопку сценарию передается строка *имя=текст кнопки*.

11) Кнопка для загрузки файлов (browse)

Служит для реализации загрузки файлов на сервер. Объект `browse` начитается с парных тегов `<form></form>`. Начинаящий тэг `<form>` содержит необходимый атрибут `enctype`. Атрибут *enctype* принимает значение *multipart/form-data*, который извещает сервер о том, что вместе с обычной информацией посылается и файл. При создании текстового поля также необходимо указать тип файла – *"file"*.

```
<form enctype="multipart/form-
data" action="upload.php" method="post">
  Загрузить файл: <input name="my_file" type="file">
  <input type="submit" value="Отправить">
</form>
```

12) Рамка (fieldset)

Объект *fieldset* позволяет вам нарисовать рамку вокруг объектов. Имеет закрывающий тэг *</fieldset>*. Заголовок указывается в тэгах *<legend></legend>*. Основное назначение объекта – задание различных стилей оформления.

Пример:

```
<fieldset>
<legend>Программное обеспечение(заголовок рамки)</legend>
Текст, который будет помещен внутри рамки.</fieldset>
```

Обработка форм

Все данные, которые вы хотите получить из HTML-формы в PHP сценарий обрабатываются с помощью суперглобальных массивов *\$_POST* или *\$_GET*, в зависимости от указанного в атрибуте *method* метода передачи данных.

1.3 Пример создания формы.

(предметная область – принтеры)

```
<html>
<head>
<title>border-spacing</title>
<style>
table {
  align="center"
  border: 0px double #0; /* Рамка вокруг таблицы */
  border-collapse: separate; /* Способ отображения границы */
  width: 70%; /* Ширина таблицы */
  border-spacing: 10px 5px; /* Расстояние между ячейками */
}
td {
  padding: 1px; /* Поля вокруг текста */
  border: 0px solid #a52a2a; /* Граница вокруг ячеек */
}
</style>
```

```
</head>
<body>
<form name="input" action="html_form_action.asp" method="get">
  <table border=1 bgcolor="#ddffdd">
    <tr>
      <td align="left">
        Товар:
        <input type="text" name="search" size="25" value="Принтер">
        <input type="submit" value="Поиск">
      </td>
      <td align="left">
        Цена:
        <input type="text" name="search" size="10" value="0">
        -
        <input type="text" name="search" size="10" value="140000">
        <input type="submit" value="Поиск">
      </td>
    </tr>
    <tr>
      <td align="left">
        Вид:
        <br>
        <select name="vid" multiple size=4>
          <option value="str">Струйный
          <option value="mat">Матричный
          <option value="laz" selected="selected">Лазерный
          <option value="3d">3D
        </select>
      </td>
      <td align="left">
        Производитель:
        <br>
        <select name="proizvod">

          <option value="Canon">Canon
          <option value="Epson">Epson
          <option value="HP">HP

          <option value="Brother">Brother
          <option value="Kyocera">Kyocera
          <option value="Pantum">Pantum
          <option value="Ricon">Ricon
          <option value="Samsung">Samsung
          <option value="Xerox">Xerox
          <option value="Cel">Cel
```

```

        <option value="Createbot">Createbot
        <option value="Myriwell">Myriwell
        <option value="UP!">UP!
        <option value="Wanhao">Wanhao
        <option value="XYZprinting">XYZprinting
        <option value="МастерКит">МастерКит
    </select>
</td>
<td align="left">
    Цветность печати:
    <br>
    <input type="checkbox" name="color">
    Цветная
    <br>
    <input type="checkbox" name="notColor">
    Чёрно-белая
</td>
</tr>
</table>
</form>
</body>
</html>

```

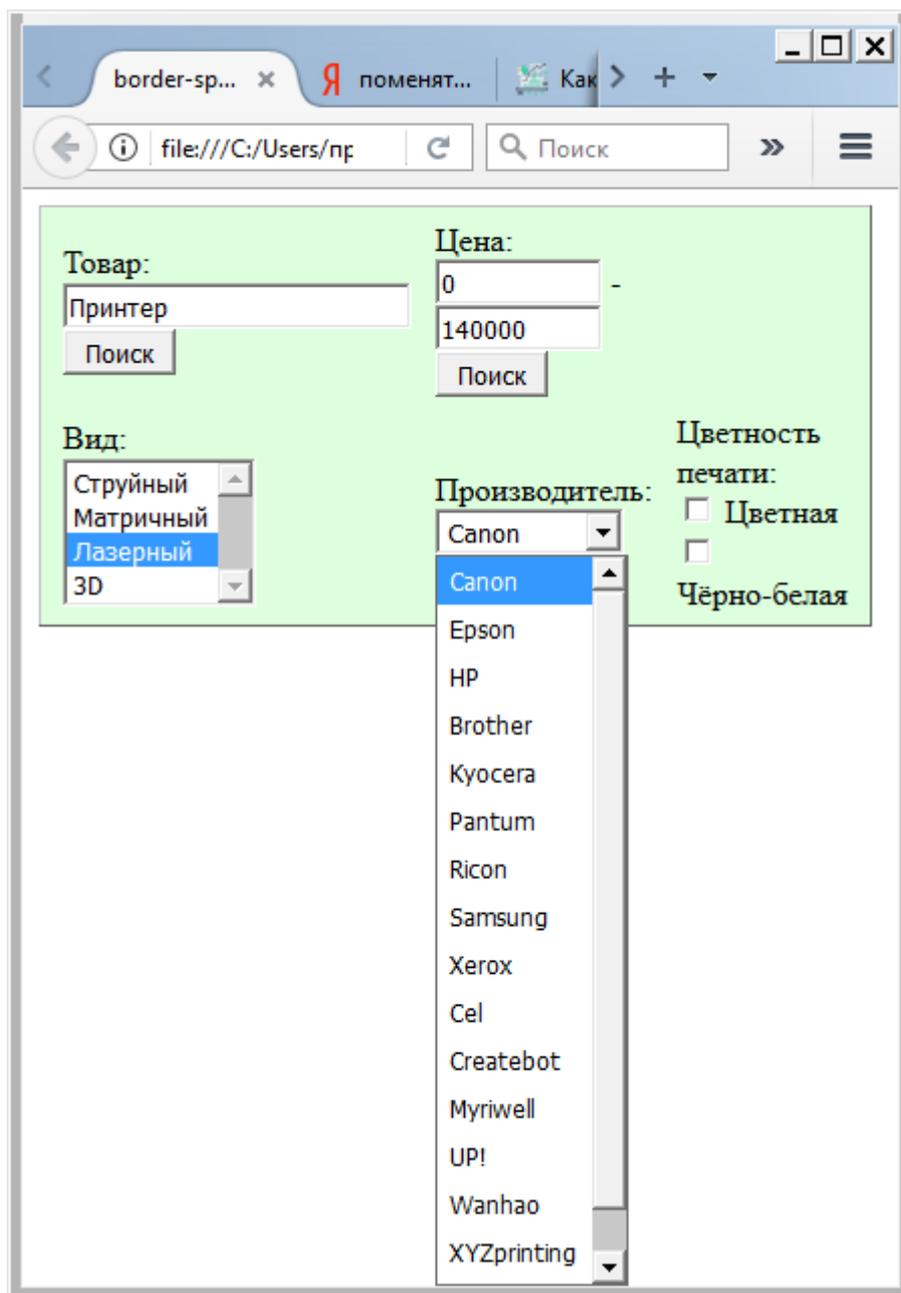
Общий вид интерфейсной формы

border-sp... x Я поменять... Как > + ▾

file:///C:/Users/np Поиск >> ≡

Товар: <input type="text" value="Принтер"/> <input type="button" value="Поиск"/>	Цена: <input type="text" value="0"/> - <input type="text" value="140000"/> <input type="button" value="Поиск"/>	
Вид: <input type="text" value="Струйный"/> <input type="text" value="Матричный"/> <input type="text" value="Лазерный"/> <input type="text" value="3D"/>	Производитель: <input type="text" value="Canon"/>	Цветность печати: <input type="checkbox"/> Цветная <input type="checkbox"/> Чёрно-белая

Выбор конкретного производителя



border-sp... x Я поменять... Как > + ▾

file:///C:/Users/np Поиск >> ≡

Товар:

Цена: -

Вид:

Производитель:

Цветность печати:
 Цветная
 Чёрно-белая

Практическая работа № 2. Создание баз данных с помощью MySQL и phpMyAdmin.

Цель: изучить создание баз данных с помощью MySQL и phpMyAdmin.

2.1 Задания.

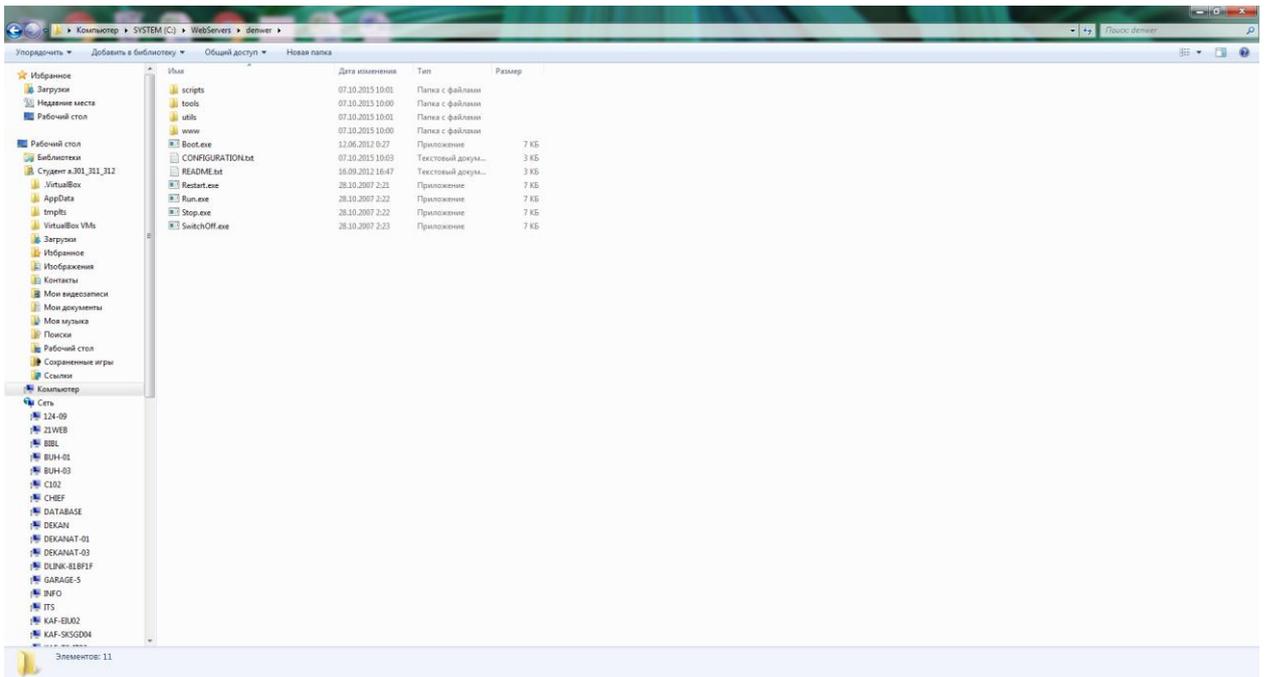
Для заданной вариантом предметной области построить базу данных средствами MySQL и phpMyAdmin. Вариант определяется в приведенной ниже таблице по последнему номеру зачетной книжки студента.

Последняя цифра в зачетной книжке студента	Номер варианта	Предметная область
1	1	Мониторы
2	2	Принтеры
3	3	Планшеты
4	4	Ноутбуки
5	5	Манипуляторы «мышь»
6	6	Центральные процессорные блоки
7	7	Смартфоны
8	8	Коммутаторы (хабы)
9	9	Клавиатуры
0	10	Сканеры

2.2 Теоритический материал.

Создание SQL базы данных.

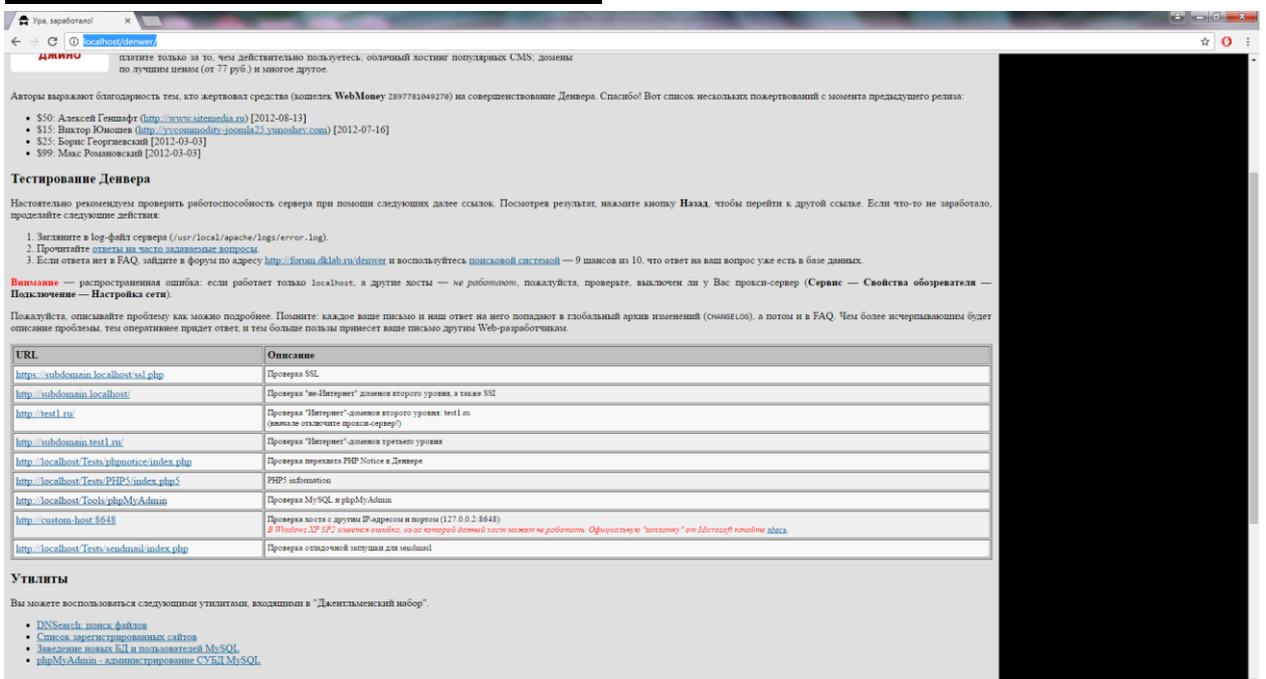
Проходим в папку C:\WebServers\denwer запускаем denwer (файл run.exe).



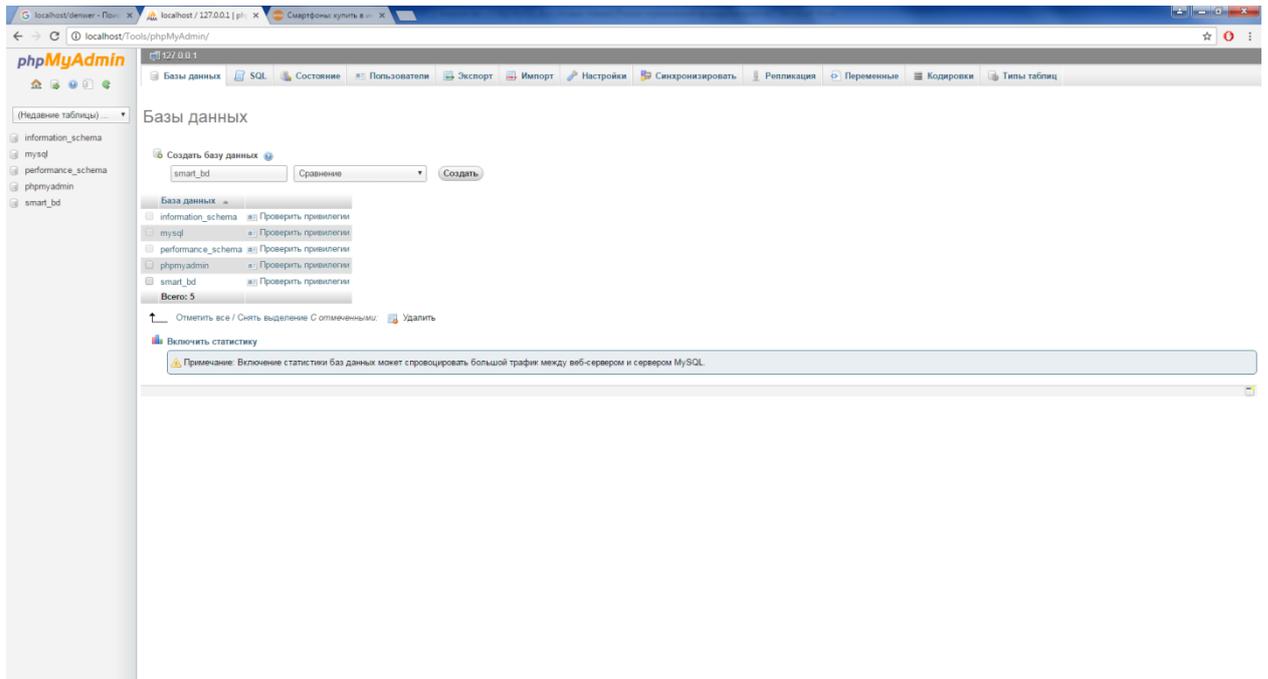
В браузере в адресной строке пишем localhost/denwer/

В открытой странице кликаем на [Проверка MySQL и phpMyAdmin](#)

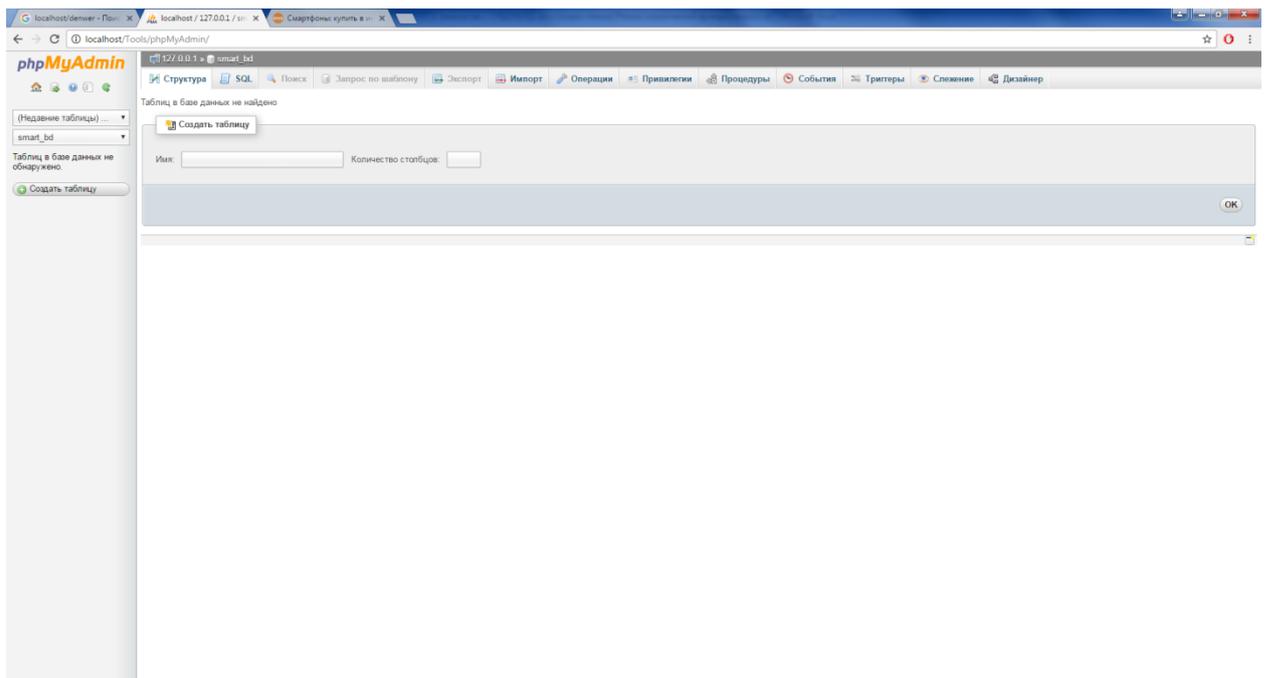
<http://localhost/Tools/phpMyAdmin>



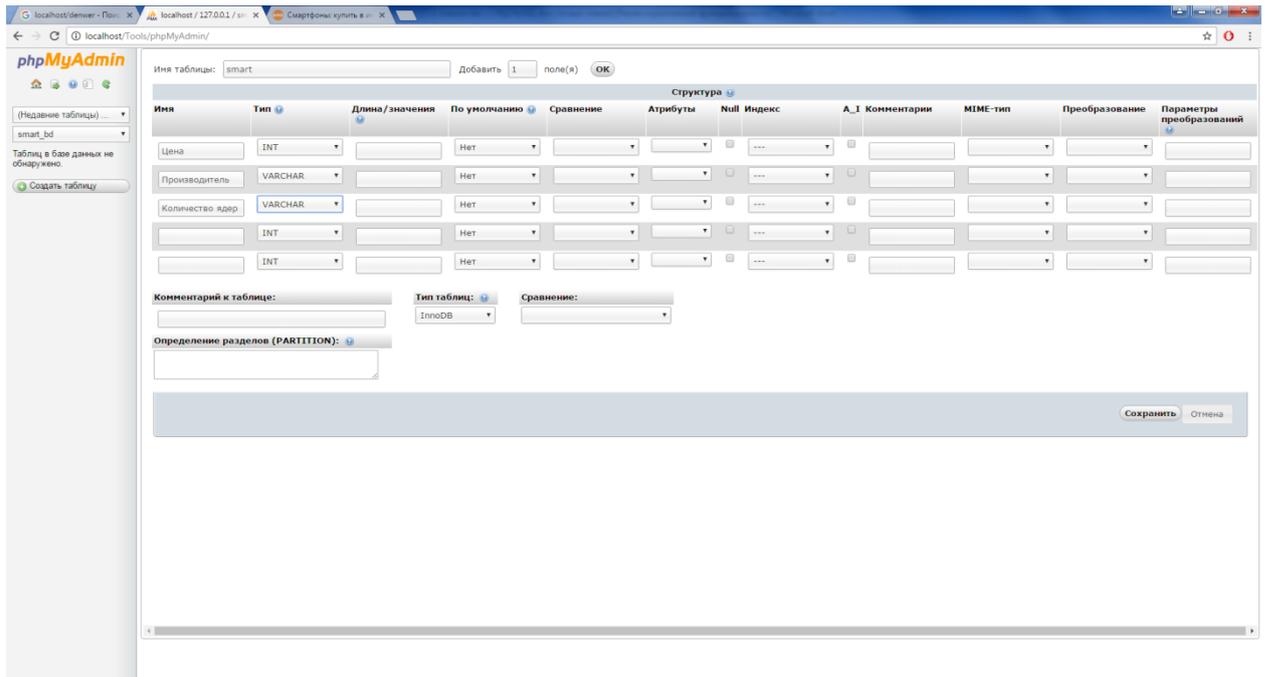
1. Создаём базу данных.



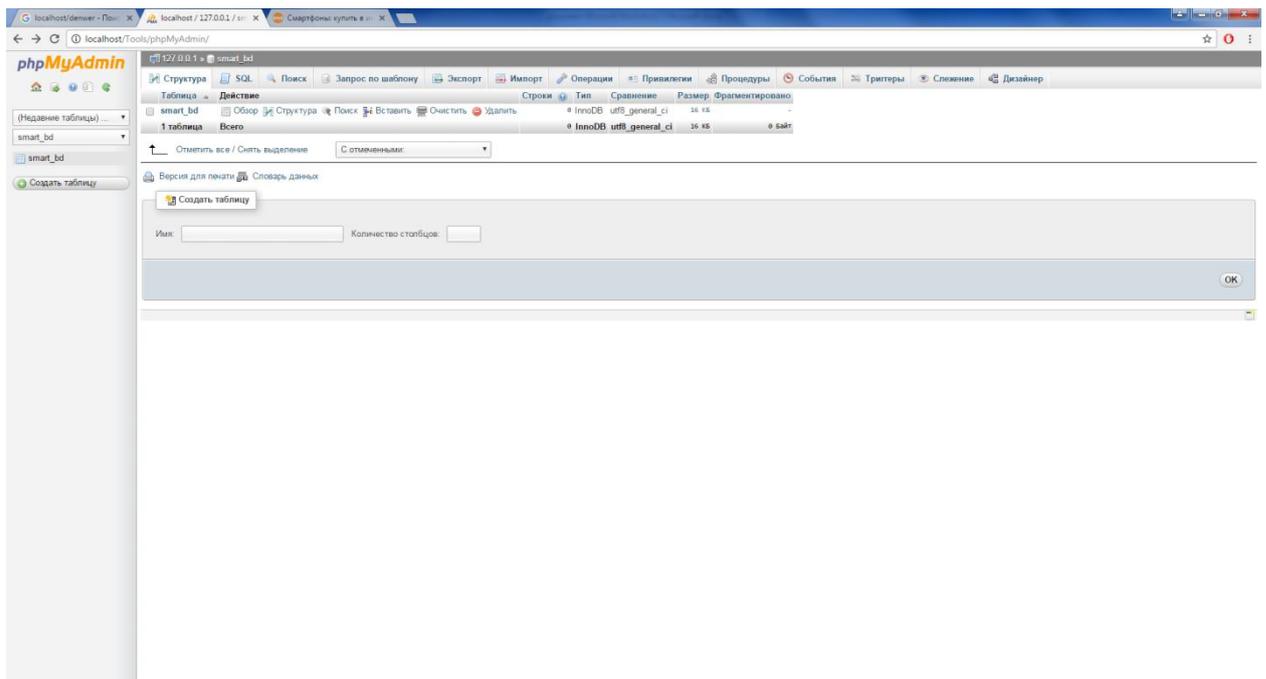
2. Создаём таблицу, в своей БД.



3. Создаём столбцы своей таблицы.



4. Таблица готова. Приступаем к её заполнению.



5. Заполняем таблицу.

The screenshot shows the 'Insert' form in phpMyAdmin. The form is divided into two sections: 'Инициировать' (Initiate) and 'Итерировать' (Iterate). The 'Инициировать' section has the following values: Price: 22900, Manufacturer: SAMSUNG, Quantity: 6, Material: металл, GPS: есть. The 'Итерировать' section has the following values: Price: 12990, Manufacturer: XIAOMI, Quantity: 8, Material: Пластик/металл, GPS: есть. Below the form, there are buttons for 'Вставить запись' (Insert record) and 'Вернуться на предыдущую страницу' (Return to previous page).

6. Таблица готова.

The screenshot shows the 'SELECT' query results for the 'smart_bd' table. The table displays columns: Price, Manufacturer, Model, Diagonal, Core, Body, and GPS. The results show 16 rows of data for various smartphone models. The first few rows are:

Price	Manufacturer	Model	Diagonal	Core	Body	GPS
6990	SENSEIT	E400 Aluminum Gray	5	4Core	пластик	нет GPS
7300	ALGATEL	One Touch POP C9 T047D POP C9 Slate	6	2Core	пластик	есть GPS
7980	VERTEX	Impress X Gold	5	4Core	пластик	есть GPS
8700	MICROSOFT	Lumia 640 Dual Sim Orange	5	2Core	пластик	есть GPS
8980	ZTE	Blade X3 4G Black	5	2Core	пластик	есть GPS
12990	XIAOMI	RedmiNote 3	6	8Core	Пластик/металл	есть GPS
13000	MEIZU	M2 mini 16Gb Gray	5	3Core	металл	есть GPS
14010	HUAWEI	P8 Lite Gold	5	8Core	металл/пластик	есть GPS
14399	LG	Class H650E Silver	5	4Core	пластик	есть GPS
15210	LENOVO	P70 Dual Sim Blue	5	8Core	пластик	есть GPS
17860	BQ	Aquaris M5 5 C000135 16Gb Black	6	8Core	металл	есть GPS
17990	ASUS	ZenPhone 3	5	4Core	Пластик	есть GPS
19990	APPLE	iPhone5s	4	2Core	металл	есть GPS
22900	SAMSUNG	GalaxyA7	6	8Core	металл	есть GPS
27980	SONY	Хрена Z3 D6603	5	4Core	металл	есть GPS
49990	SAMSUNG	Galaxy S7 SM-G930FD 32Gb DS Black Опух	5	8Core	металл/пластик	есть GPS

7. Добавляем администратора.

The screenshot displays the phpMyAdmin interface for a MySQL database named 'smart_bd'. The main content area shows the execution of SQL commands to create a new user and grant privileges:

```
CREATE USER 'kholmatov'@'localhost' IDENTIFIED BY '*****'  
GRANT USAGE ON *.* TO 'kholmatov'@'localhost' IDENTIFIED BY '*****' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0  
GRANT ALL PRIVILEGES ON 'smart_bd'.* TO 'kholmatov'@'localhost'
```

A yellow notification box with a green checkmark indicates: "Был добавлен новый пользователь." (A new user was added).

Below the notification, a table lists the users with access to the 'smart_bd' database:

Пользователь	Хост	Тип	Привилегии	GRANT	Действие
kholmatov	localhost	Глобальный символ	smart_bd ALL PRIVILEGES	Нет	Редактирование привилегий
root	127.0.0.1	Глобальный уровень	ALL PRIVILEGES	Да	Редактирование привилегий
root	:::1	Глобальный уровень	ALL PRIVILEGES	Да	Редактирование привилегий
root	localhost	Глобальный уровень	ALL PRIVILEGES	Да	Редактирование привилегий

At the bottom, there is a 'Новый' (New) button and a link to 'Добавить пользователя' (Add user).

Практическая работа № 3. Формирование запросов к базе данных SQL.

Цель: изучить формирование запросов к базе данных SQL.

3.1 Задания.

Для заданной вариантом предметной области и базы данных, построенной в практической работе №2 сформировать все запросы, рассмотренные ниже в теоретическом материале. Вариант определяется в приведенной ниже таблице по последнему номеру зачетной книжки студента.

Последняя цифра в зачетной книжке студента	Номер варианта	Предметная область
1	1	Мониторы
2	2	Принтеры
3	3	Планшеты
4	4	Ноутбуки
5	5	Манипуляторы «мышь»
6	6	Центральные процессорные блоки
7	7	Смартфоны
8	8	Коммутаторы (хабы)
9	9	Клавиатуры
0	10	Сканеры

3.2 Теоритический материал.

Запросы к базе данных SQL.

Поиск по запросу с помощью WHERE='\$sname''

```
index.php
1 <html>
2 Приветствуем вас в нашем интернет-магазине!
3 <br>
4 Выберите нужный товар с помощью поиска:
5 <br>
6 <br>
7 <form action="index.php" method="post" enctype="multipart/form-data">
8 Слово для поиска: <input type="text" name="sname" /><br />
9
10 <input type="submit" value="Искать" />
11 </form>
12
13 <br>
14 </html>
15
16
17 <?php
18 echo "";
19 $db = mysqli_connect("localhost","root","");
20 //соединение с сервером_half_compiler
21 mysqli_select_db("smart_bd",$db); //забор БД
22 mysqli_query("SET NAMES cp1251");//кодировка
23 //временная переменная переданный запрос
24 $sname=$_POST['sname'];
25 //$sname=$_POST['sname'];
26 //временная переменная sql запрос
27 $sql = "SELECT * FROM smart_bd WHERE Manufacturer='{$sname}'";
28 // $sql = "SELECT * FROM smart_bd ORDER BY GPS";
29 // $sql = "SELECT * FROM smart_bd WHERE diagonal='5'";
30 // $sql = "SELECT * FROM smart_bd WHERE Manufacturer LIKE 'S%'";
31 $result = mysqli_query($sql);//посылаем запрос mysqli
32 echo "Находим выборку";
33 echo "<br>";
34 while($item = mysqli_fetch_assoc($result))//получаем элемент выборки
35 { echo "<li> $item[Manufacturer] $item[Model] $item[diagonal] $item[Core] $item[Body] $item[GPS] $item[Price]</li>"; //вывод результатов на экран
36 echo "<br>";
37 //Добавить строку в таблицу
38
39
40 mysqli_close($db);// закрытие соединения с БД и сервером
41
}
```

PHP Hypertext Preprocessor file

length:1610 lines:41

Ln:27 Col:1 Sel:0|0

Dev:Windows

Windows-1251

ANS



Сортировка с помощью ORDER BY

```
index.php
1 <html>
2 Приветствуем вас в нашем интернет-магазине!
3 <br>
4 Выберите нужный товар с помощью поиска:
5 <br>
6 <br>
7 <form action="index.php" method="post" enctype="multipart/form-data">
8 Слово для поиска: <input type="text" name="sname" /><br />
9
10 <input type="submit" value="Искать" />
11 </form>
12
13 <br>
14 </html>
15
16
17 <?php
18 echo "";
19 $db = mysqli_connect("localhost","root","");
20 //соединение с сервером_half_compiler
21 mysqli_select_db("smart_bd",$db); //забор БД
22 mysqli_query("SET NAMES cp1251");//кодировка
23 //временная переменная переданный запрос
24 $sname=$_POST['sname'];
25 // $sname=$_POST['sname'];
26 //временная переменная sql запрос
27 $sql = "SELECT * FROM smart_bd WHERE GPS='{$sname}'";
28 $result = mysqli_query($sql);//посылаем запрос mysqli
29 echo "Находим выборку";
30 echo "<br>";
31 while($item = mysqli_fetch_assoc($result))//получаем элемент выборки
32 { echo "<li> $item[Manufacturer] $item[Model] $item[diagonal] $item[Core] $item[Body] $item[GPS] $item[Price]</li>"; //вывод результатов на экран
33 echo "<br>";
34 $sql = "SELECT * FROM smart_bd ORDER BY Manufacturer='{$sname}'";
35 $result = mysqli_query($sql);//посылаем запрос mysqli
36 echo "Находим выборку";
37 echo "<br>";
38 while($item = mysqli_fetch_assoc($result))//получаем элемент выборки
39 { echo "<li> $item[Manufacturer] $item[Model] $item[diagonal] $item[Core] $item[Body] $item[GPS] $item[Price]</li>"; //вывод результатов на экран
40 echo "<br>";
41 //Добавить строку в таблицу
42
43
44 mysqli_close($db);// закрытие соединения с БД и сервером
45
}
```

PHP Hypertext Preprocessor file

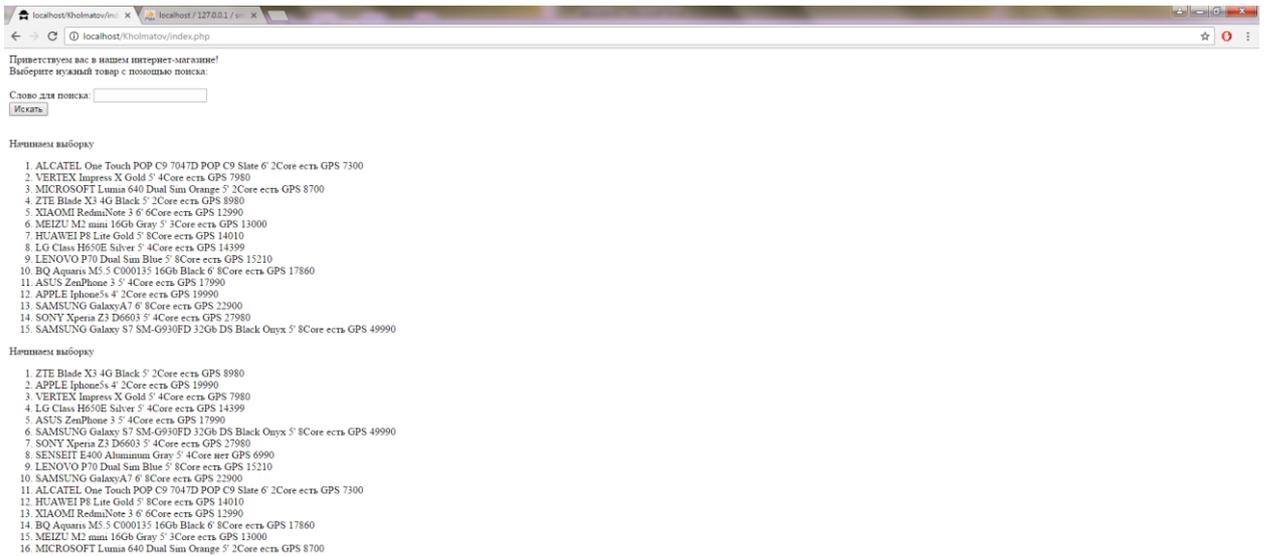
length:1910 lines:45

Ln:34 Col:54 Sel:0|0

Dev:Windows

Windows-1251

ANS

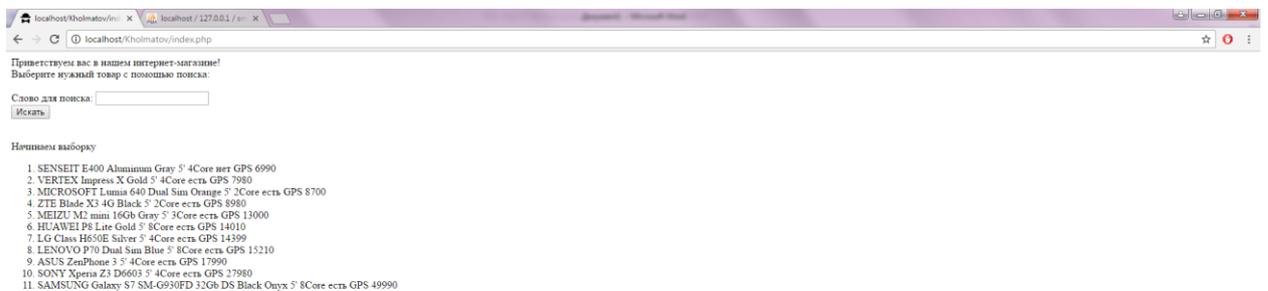


Вставка условия **WHERE**

```

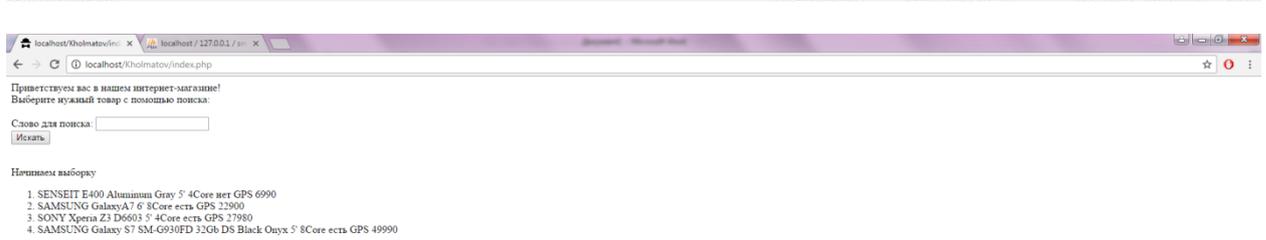
1 <html>
2 Приветствуем вас в нашем интернет-магазине!
3 <br>
4 Выберите нужный товар с помощью поиска:
5 <br>
6 <br>
7 <form action="index.php" method="post" enctype="multipart/form-data">
8 Слово для поиска: <input type="text" name="aname" /><br />
9
10 <input type="submit" value="Искать" />
11 </form>
12
13 <br>
14 </html>
15
16
17 <?php
18 echo "<br>";
19 $db = mysqli_connect("localhost","root","");
20 //соединение с сервером halt_compiler
21 mysqli_select_db("smart_bd",$db); //выбор БД
22 mysqli_query("SET NAMES utf8"); //кодировка
23 //переменная хранения переданный запрос
24 $aname=$_POST['aname'];
25 //формальная переменная sql запрос
26 //формальная переменная sql запрос
27 //sql = "SELECT * FROM smart_bd WHERE Manufacturer='".$aname"."";
28 //sql = "SELECT * FROM smart_bd ORDER BY GPS";
29 $sql = "SELECT * FROM smart_bd WHERE diagonal='5'";
30 $result = mysqli_query($sql); //выполнение запроса mysqli
31 echo "Начинаем выборку";
32 echo "<br>";
33 while($row = mysqli_fetch_assoc($result)) //получаем элемент выборки
34 { echo "<table border='1'> <tr><td>Manufacturer: <td>Model: <td>diagonal: <td>Core: <td>Body: <td>Price: </tr>"; //вывод результатов на экран
35 echo "</td>";
36 //Добавить строку в таблицу
37
38 }
39 mysqli_close($db); // закрытие соединения с БД и сервером
40

```



Поиск Запроса по букве с помощью **LIKE**

```
Z:\home\localhost\www\kholmatov\index.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
index.php
1 <html>
2 Приветствуем вас в нашем интернет-магазине!
3 <br>
4 Выберите нужный товар с помощью поиска:
5 <br>
6 <br>
7 <form action="index.php" method="post" enctype="multipart/form-data">
8 Слово для поиска: <input type="text" name="sname" /><br />
9
10 <input type="submit" value="Искать" />
11 </form>
12
13 <br>
14 </html>
15
16
17 <?php
18 echo "";
19 $db = mysql_connect("localhost","root","");
20 //соединение с сервером halt_compiler
21 mysql_select_db("smart_bd",$db); //выбор БД
22 mysql_query("SET NAMES cp1251");//кодировка
23 //переменная хранящая переданный запрос
24 $sname=$_POST['sname'];
25 //переменная хранящая sql запрос
26 //sql = "SELECT * FROM smart_bd WHERE Manufacturer='$sname'";
27 //sql = "SELECT * FROM smart_bd ORDER BY GPS";
28 //sql = "SELECT * FROM smart_bd WHERE diagonal=5 ";
29 $sql = "SELECT * FROM smart_bd WHERE Manufacturer LIKE '%$s%'";
30 $result = mysql_query($sql);//посылает запрос mysql
31 echo "Находим выборку";
32 echo "<br>";
33 while($item = mysql_fetch_assoc($result))//получение элемента выборки
34 { echo "<table border='1'>";
35 <tr>
36 <td> $item[Manufacturer] $item[Model] $item[diagonal] $item[Core] $item[Body] $item[GPS] $item[Price]</td> //вывод результатов на экран
37 </tr>
38 </table>";
39 //добавить строку в таблицу
40 }
41 mysql_close($db); // закрыть соединение с БД и сервером
```



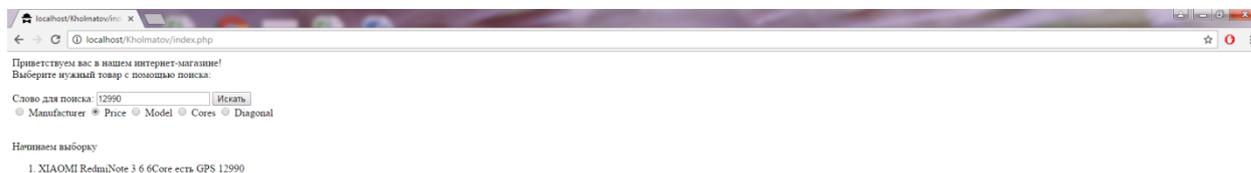
Вставка “radio”

```
Z:\home\localhost\www\kholmatov\index.php - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
index.php
1 <html>
2 Приветствуем вас в нашем интернет-магазине!
3 <br>
4 Выберите нужный товар с помощью поиска:
5 <br>
6 <br>
7 <form action="index.php" method="post" enctype="multipart/form-data">
8 Слово для поиска: <input type="text" name="sname" /> <input type="submit" value="Искать" /> <br />
9 <input name="name" type="radio" value="1" /> Manufacturer:
10 <input name="name" type="radio" value="2" /> Price
11 <input name="name" type="radio" value="3" /> Model
12 <input name="name" type="radio" value="4" /> Cores
13 <input name="name" type="radio" value="5" /> Diagonal
14
15
16 </form>
17
18 <br>
19 </html>
20
21
22 <?php
23 echo "";
24 $db = mysql_connect("localhost","root","");
25 //соединение с сервером halt_compiler
26 mysql_select_db("smart_bd",$db); //выбор БД
27 mysql_query("SET NAMES cp1251");//кодировка
28 //переменная хранящая переданный запрос
29 $sname=$_POST['sname'];
30 $name=$_POST['name'];
```

Вставка "if"

```
23 echo " ";
24 $db = mysql_connect("localhost", "root", "");
25 //соединение с сервером halt_compiler
26 mysql_select_db("smart_bd", $db); //выбор БД
27 mysql_query('SET NAMES cp1251');//кодировка
28 //переменная хранящая переданный запрос
29 $sname=$_POST['sname'];
30 $sname=$_POST['name'];
31 //переменная хранящая sql запрос
32 if($sname==1){$sql = "SELECT * FROM smart_bd WHERE Manufacturer='$sname'";}
33 if($sname==2){$sql = "SELECT * FROM smart_bd WHERE Price='$sname'";}
34 if($sname==3){$sql = "SELECT * FROM smart_bd WHERE Model='$sname'";}
35 if($sname==4){$sql = "SELECT * FROM smart_bd WHERE Cores='$sname'";}
36 if($sname==5){$sql = "SELECT * FROM smart_bd WHERE Diagonal='$sname'";}
37 // $sql = "SELECT * FROM smart_bd WHERE Manufacturer='$sname'";
38 // $sql = "SELECT * FROM smart_bd ORDER BY GPS";
39 // $sql = "SELECT * FROM smart_bd WHERE diagonal=5 ";
40 // $sql = "SELECT * FROM smart_bd WHERE Manufacturer LIKE 'S%'";
41 $result = mysql_query($sql);//посылаем запрос mysql
42 echo "Начинаем выборку";
43 echo "<ol>";
44 while($item = mysql_fetch_assoc($result))//получения элемента выборки
45 { echo "<li> $item[Manufacturer] $item[Model] $item[Diagonal] $item[Cores] $item[Body] $item[GPS] $item[Price]</li>"; } //ВЫЕ
46 echo "</ol>";
47 //Добавить строку в таблицу
48
49
50 mysql_close($db);// закрытие соединения с БД и сервером
51
```

Получаем результат: Выполняя **поиск по цене(Price)**, нам выдаёт нужный результат. И аналогично происходит поиск по другим параметрам.



Практическая работа № 4. Изучение среды разработки Web приложений.

Цель: изучить возможности среды разработки Web приложений CodeLobster.

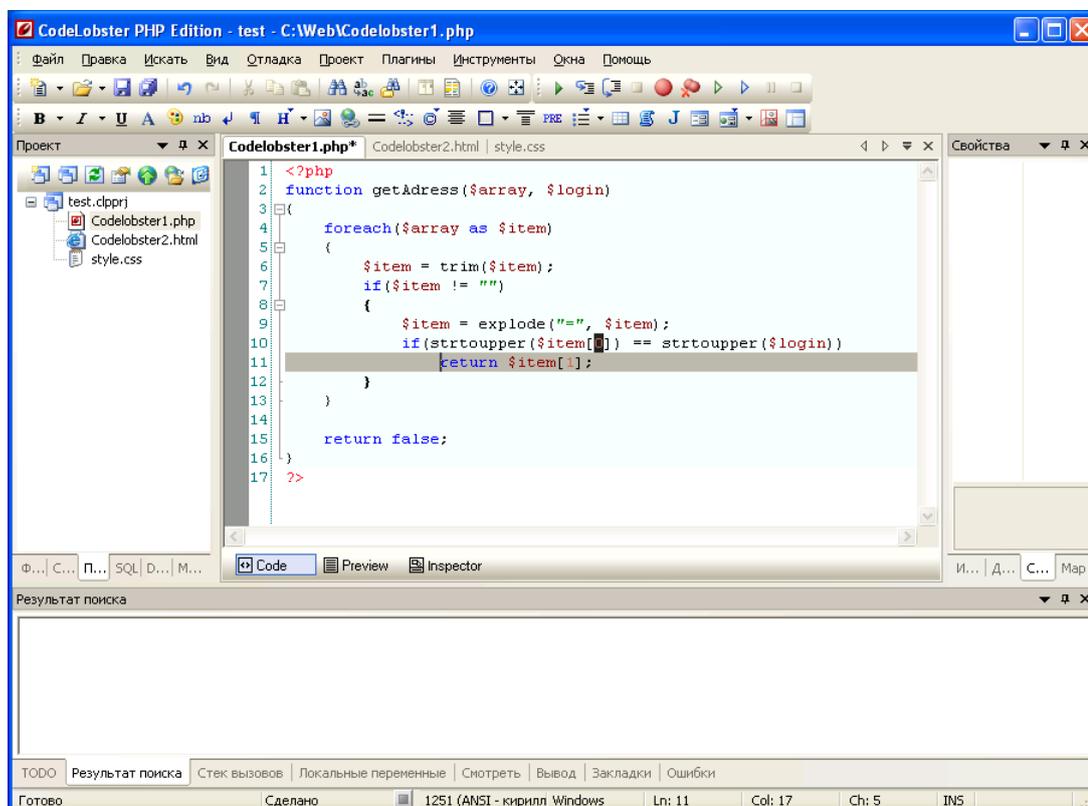
4.1 Задания.

Изучить возможности среды разработки Web приложений CodeLobster. Для программы интерфейсной формы, разработанной в практической работе №1 построить Web приложение в среде CodeLobster.

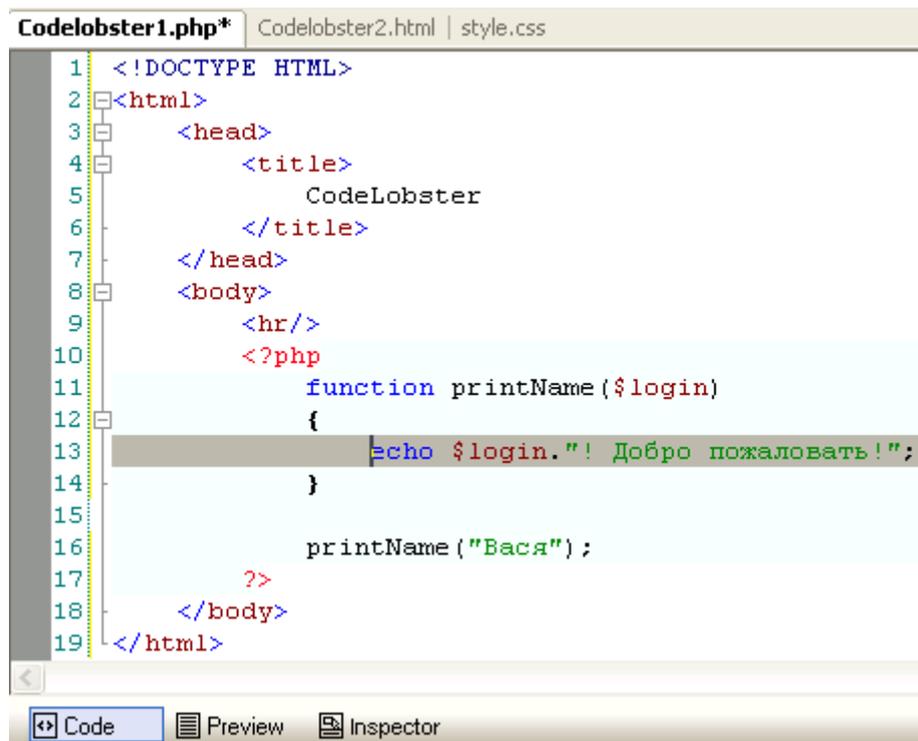
4.2 Теоритический материал.

Среды разработки Web приложений. Программа CodeLobster.

Интерфейс программы имеет стандартную для большинства IDE структуру панелей с набором вкладок. Расположение панелей легко настраивается простым перетаскиванием мышью. Каждая панель содержит определенное число вкладок: карта файла, структура класса, свойства, ошибки, стек вызова, менеджер проекта и др. Вкладки можно скрывать, добавлять и менять местами. В верхней части располагаются панели инструментов.

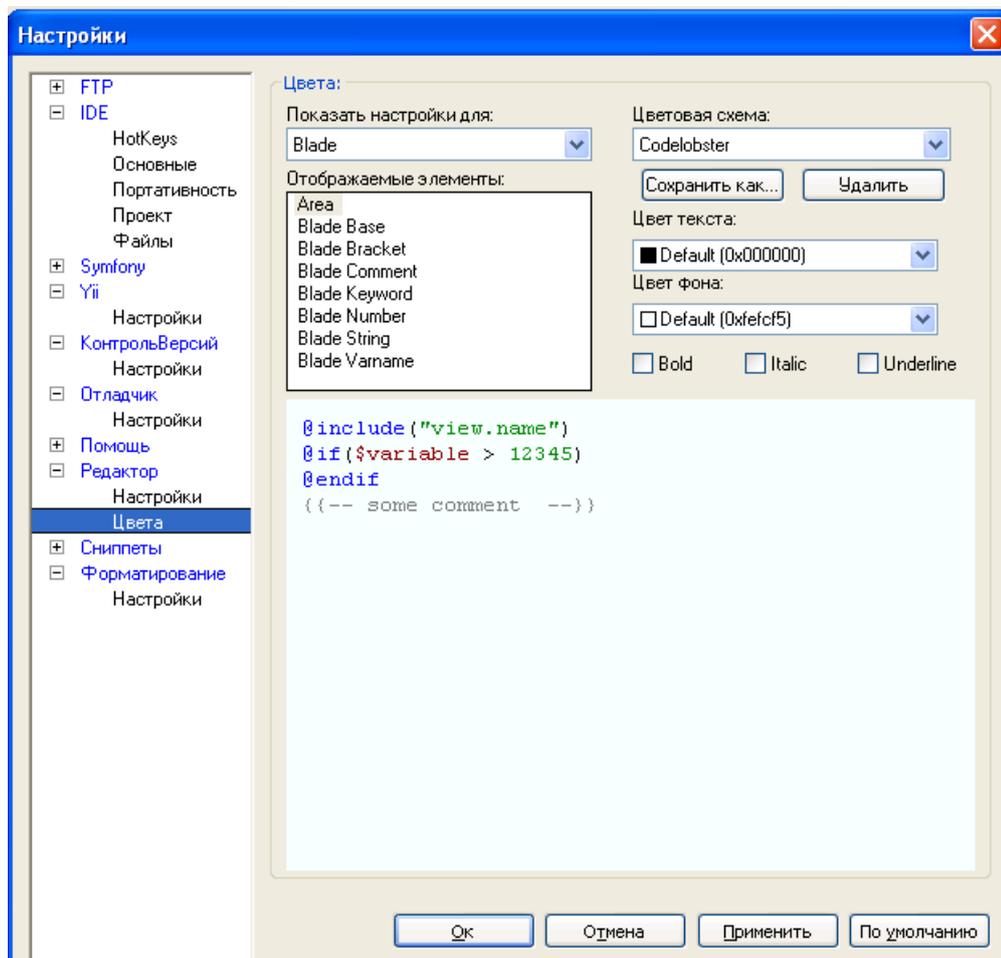


1. **Подсветка синтаксиса** самых популярных языков для web-разработки: PHP, HTML, CSS, JavaScript. Подсветка кода также прекрасно работает и в файлах со смешанным кодом:

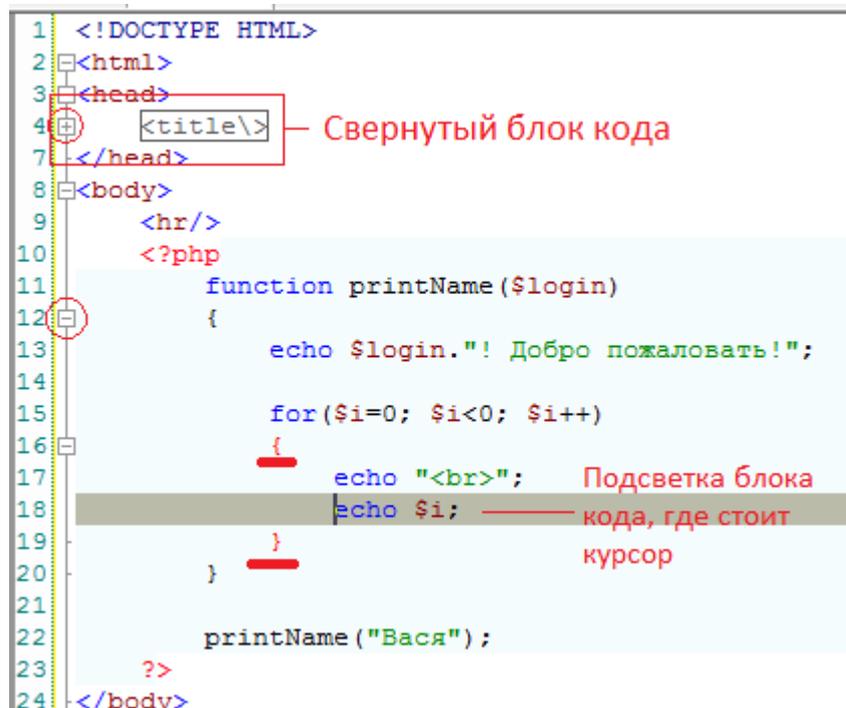


```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>
5       CodeLobster
6     </title>
7   </head>
8   <body>
9     <hr/>
10    <?php
11      function printName ($login)
12      {
13        echo $login."! Добро пожаловать!";
14      }
15
16      printName ("Вася");
17    ?>
18  </body>
19 </html>
```

Каким цветом подсвечивать теги, ключевые слова, атрибуты, названия функций и др. можно настроить на свое усмотрение в разделе "Редактор" -> "Цвета" окна "Настройки", которое можно вызвать через меню "Инструменты" -> "Настройки...":



Кроме подсветки синтаксиса, CodeLobster умеет подсвечивать парные теги и скобки. Например, в PHP блоки кода обычно заключаются в фигурные скобки и установив курсор внутри такого блока редактор подсвечивает соседние скобки. То же самое происходит и с парными тегами в HTML коде. Такие блоки кода можно сворачивать и разворачивать с помощью кнопок минус и плюс:



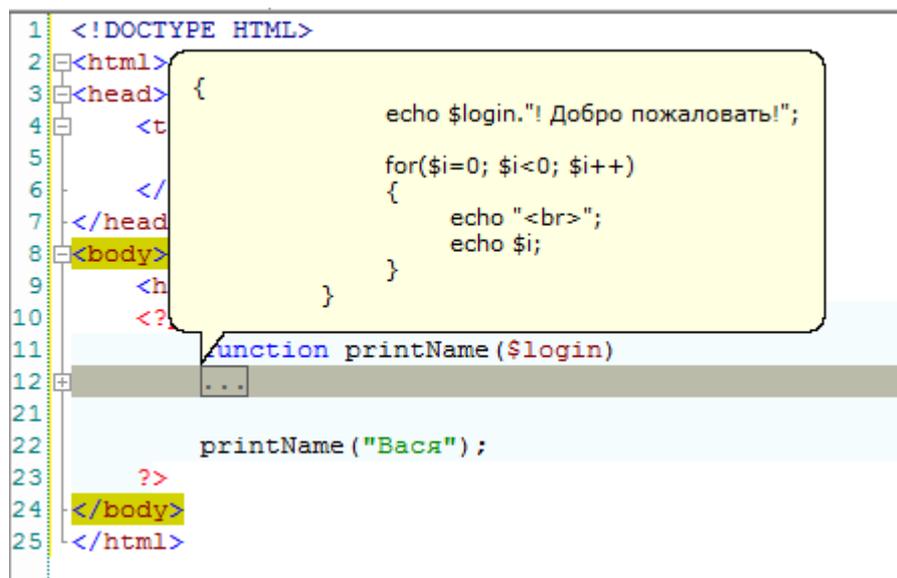
The screenshot shows a code editor with the following code:

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <title>\</title>
7 </head>
8 <body>
9 <hr/>
10 <?php
11 function printName ($login)
12 {
13     echo $login."! Добро пожаловать!";
14
15     for($i=0; $i<0; $i++)
16     {
17         echo "<br>";
18         echo $i;
19     }
20
21     printName ("Вася");
22
23 >?<
24 </body>
```

Annotations in the image:

- A red box around lines 4-7 is labeled "Свернутый блок кода" (Collapsed code block).
- A red circle around the opening curly brace on line 12 is labeled "Подсветка блока кода, где стоит курсор" (Code block highlighting where the cursor is).

Если навести курсор на свернутый блок кода, то всплывающая подсказка в виде баллона покажет его содержимое:



The screenshot shows the same code as the previous image, but with a tooltip (yellow box) over the collapsed block on line 4. The tooltip contains the following code:

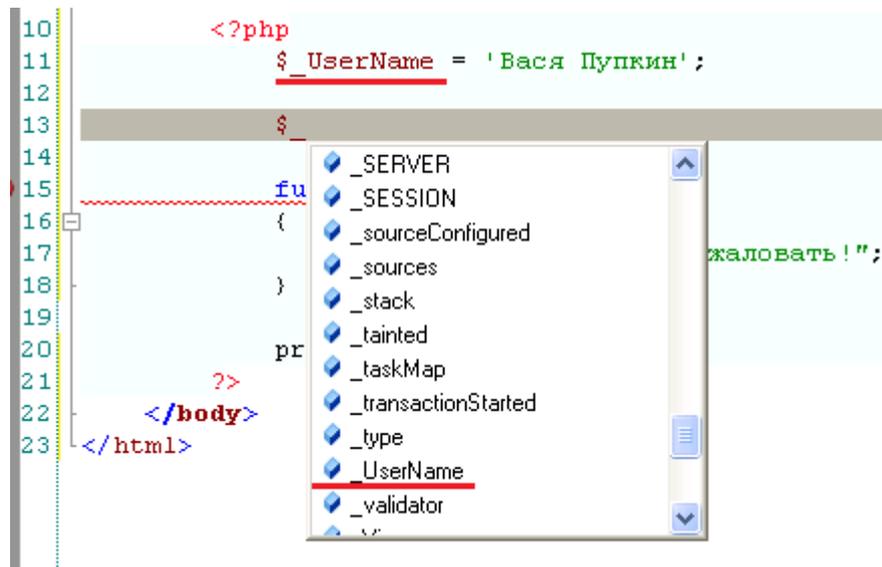
```
{
    echo $login."! Добро пожаловать!";
    for($i=0; $i<0; $i++)
    {
        echo "<br>";
        echo $i;
    }
}
```

The code in the editor below the tooltip is partially visible, showing the function definition and call.

Сворачивание блоков кода, выделение блоков кода, всплывающая подсказка — все эти функции упрощают работу с файлами содержащими большое количество строк.

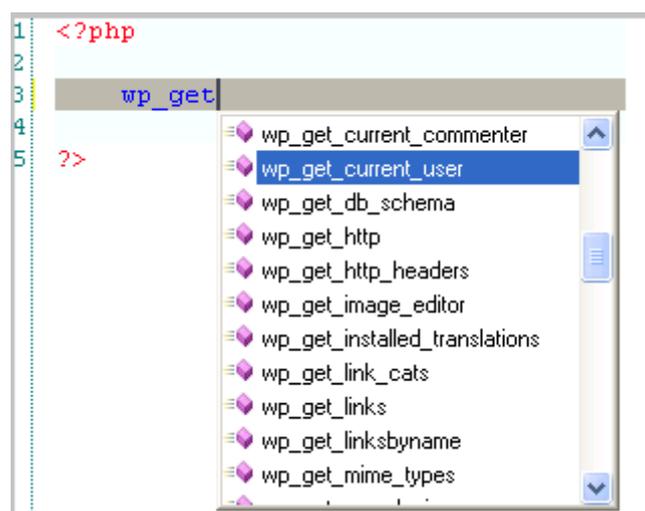
2. В редакторе мощная функция **автодополнения** для языков PHP, HTML, CSS, JavaScript. Т.е. вы начинаете набирать имя функции, тег, CSS-свойство и др. и сразу же появляется всплывающий список позволяющий быстро выбрать нужный вариант.

Причем для языков PHP и JavaScript всплывающий список содержит не только встроенные константы и функции, но и пользовательские функции, классы и переменные:



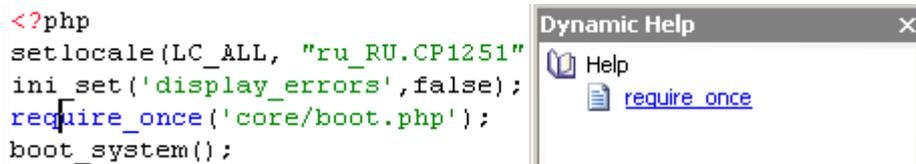
Для языка HTML кроме автодополнения тегов, атрибутов и значений атрибутов, также автоматически вставляются закрывающие теги.

Если у вас профессиональная версия, то можно включить автодополнение для популярных CMS и PHP-фреймворков, таких как Joomla, WordPress, Drupal, CakePHP, Yii и др. Например, автодополнение для функций и аргументов WordPress :

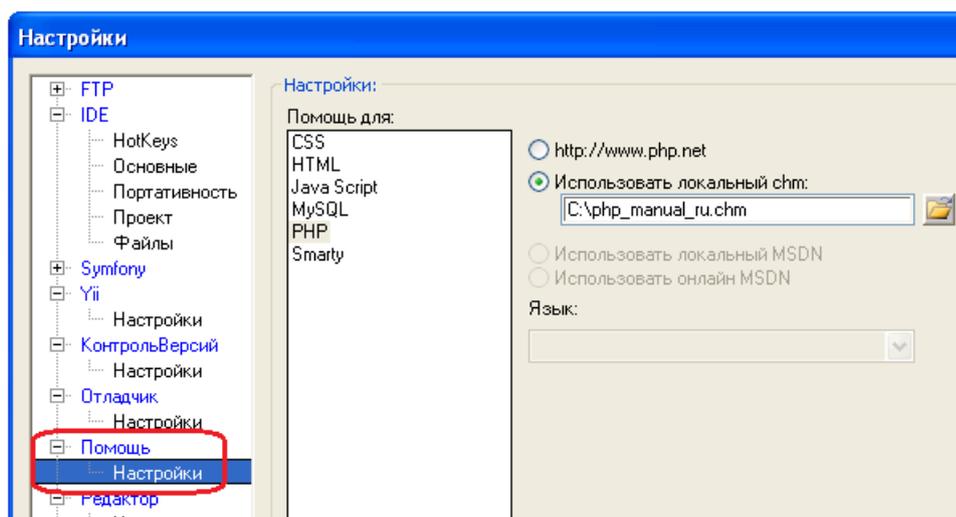


С функцией автодополнения написание кода в CodeLobster PHP Editor значительно ускоряется.

3. **Контекстная справка** по всем поддерживаемым языкам. Устанавливаете курсор на функцию, атрибут или тег и при нажатии F1 появится список ссылок, щелкнув по одной из которых, на странице откроется подробное описание по текущему элементу с официального сайта при наличии интернета:

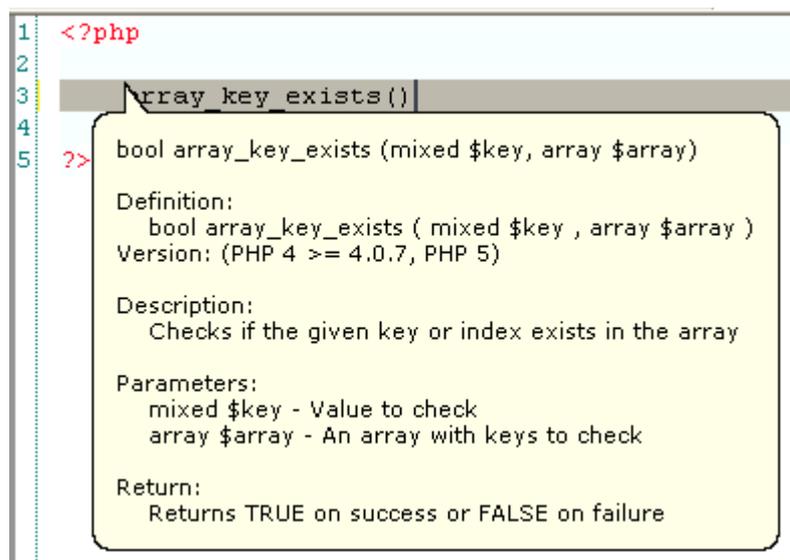


Если интернета нет, то описание открывается с локальной справочной системы (MSDN или файл справки формата chm), которую только необходимо предварительно скачать и настроить:



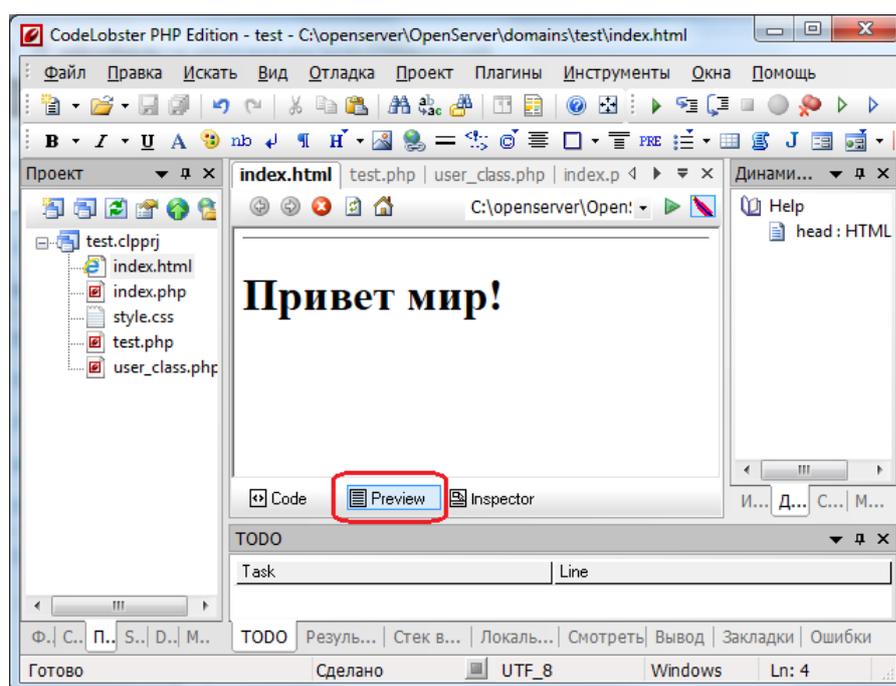
При установке соответствующих плагинов контекстная справка также работает и для следующих CMS и фреймворков: AngularJS, CodeIgniter, Drupal API, Facebook API, JQuery, Joomla, Smarty, Twig, WordPress и Yii.

Кроме контекстной справки, при наведении курсора на функцию всплывает баллон с краткий описанием функции (на английском языке), что также очень удобно, если, например, точно не помнишь какие аргументы принимает функция или, что она возвращает:



4. **Инспектор кода.** Редактор Codelobster PHP Edition имеет три режима работы:

- **Code** — собственно набор и редактирование кода.
- **Preview** — предпросмотр результата работы кода, т.е. сразу в окне редактора вы можете увидеть результат работы кода, нажав закладку «preview»:



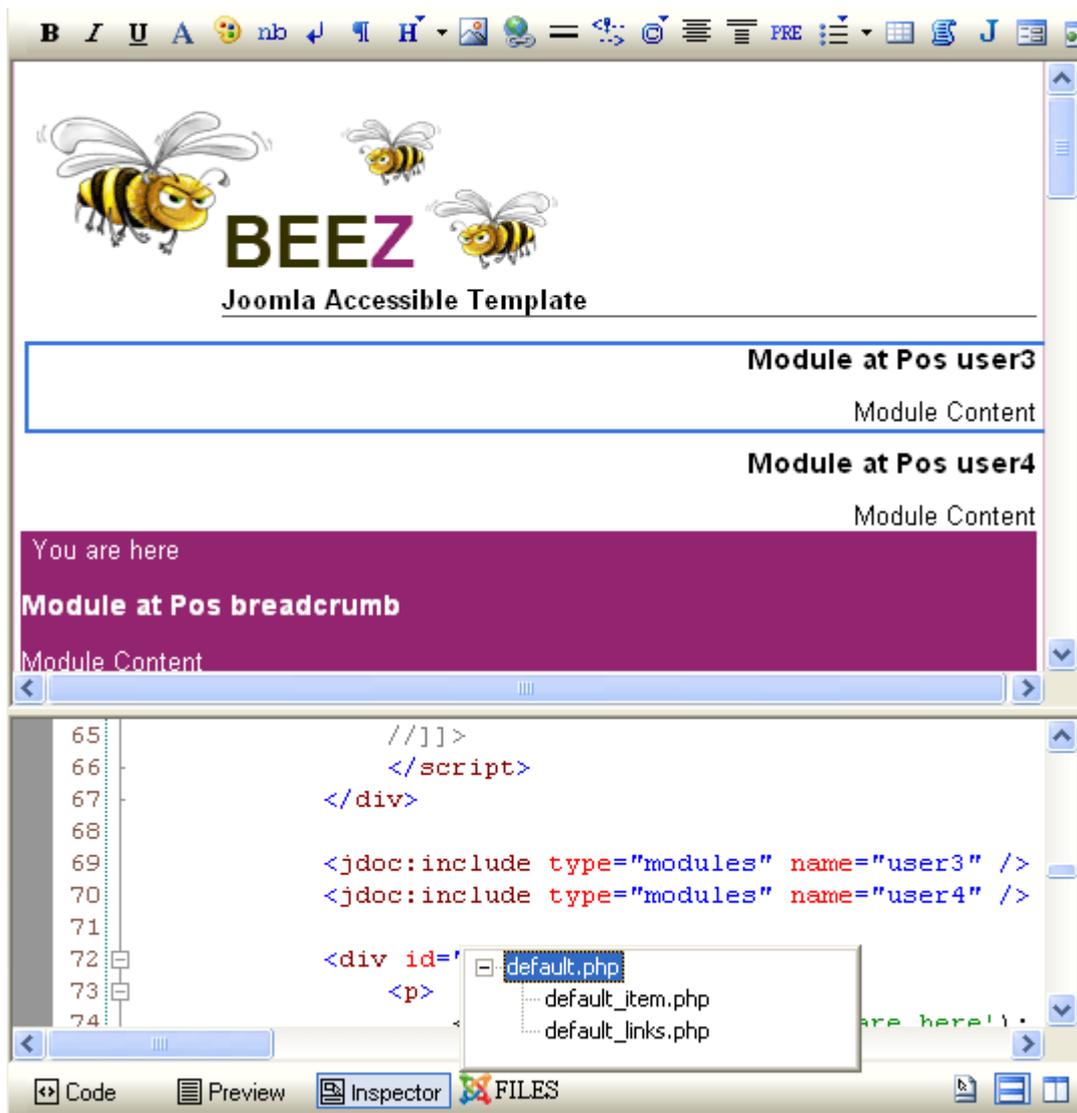
- **inspector** — так называемый **инспектор кода**, достаточно полезный режим работы с кодом. В этом режиме окно редактора делится на три области:



В верхней области отображается результат работы кода, в нижней левой области происходит редактирование кода, а правой нижней отображаются CSS стили элементов веб страницы.

В этом режиме очень удобно работать с HTML кодом. Сильно упрощается поиск нужных участков кода, когда выделив нужный элемент в области предпросмотре сразу же отображается его код в левой области и правила CSS-стилей влияющих на него в правой области.

Режим инспектора также очень удобен при работе над темами для движков WordPress и Joomla:



5. Наличие **php отладчика**, который позволяет отладить приложение перед загрузкой на веб-сервер.

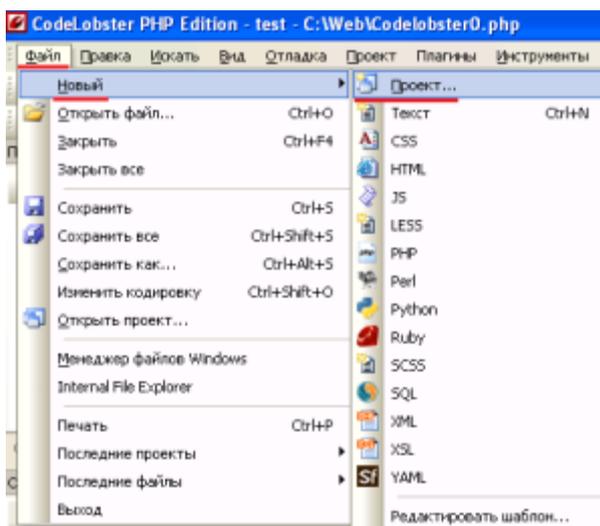
Работа с отладчиком ведется с помощью соответствующей панели инструментов:



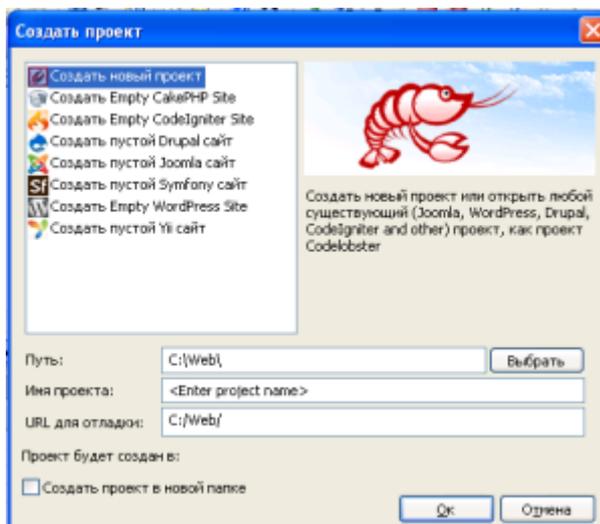
С помощью панели инструментов можно запустить отладку, расставить точки остановки приложения и выполнить каждую строку кода по шагам.

Запустив отладку, приложение будет останавливать свое выполнение в местах, где расставлены точки остановки. Когда приложение остановилось появляется возможность проверить значение переменных в этом месте выполнения программы, посмотреть стек вызова функций и продолжить его выполнение по шагам.

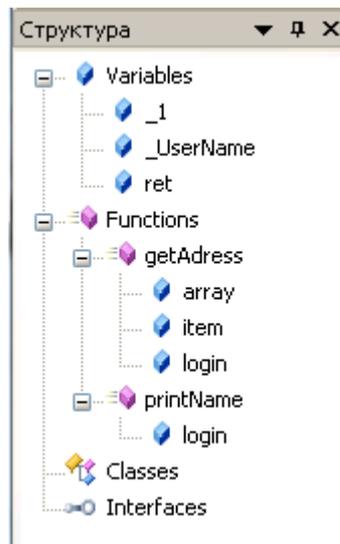
6. Есть возможность **объединить файлы в проект**. Для этого идете в меню "Файл" -> "Новый" -> "Проект":



Откроется окно создания проекта:



Можно создать простой PHP проект, а при наличии плагинов можно создать сайта с помощью CakePHP, CodeIgniter, Drupal, Joomla, Symfony, WordPress и Yii. В проекте можно создавать новые файлы или добавлять существующие. Объединив файлы в проект, легко осуществлять поиск нужных переменных, функций и классов, так как они группируются в структуру проекта:



При настроенном FTP соединении можно сразу же загружать проект на веб-сервер. Также можно подключить к проекту систему контроля версий типа GIT, SVN и др.

Из остальных возможностей можно отметить навигацию на описание функций и подключаемые файлы при нажатой клавише CTRL, превью в браузере, закладки, portable версия, всплывающая подсказка показывающее изображение и ее размер:

Практическая работа № 5. Каскадные таблицы стилей (CSS).

Цель: изучить формирование каскадных таблиц стилей.

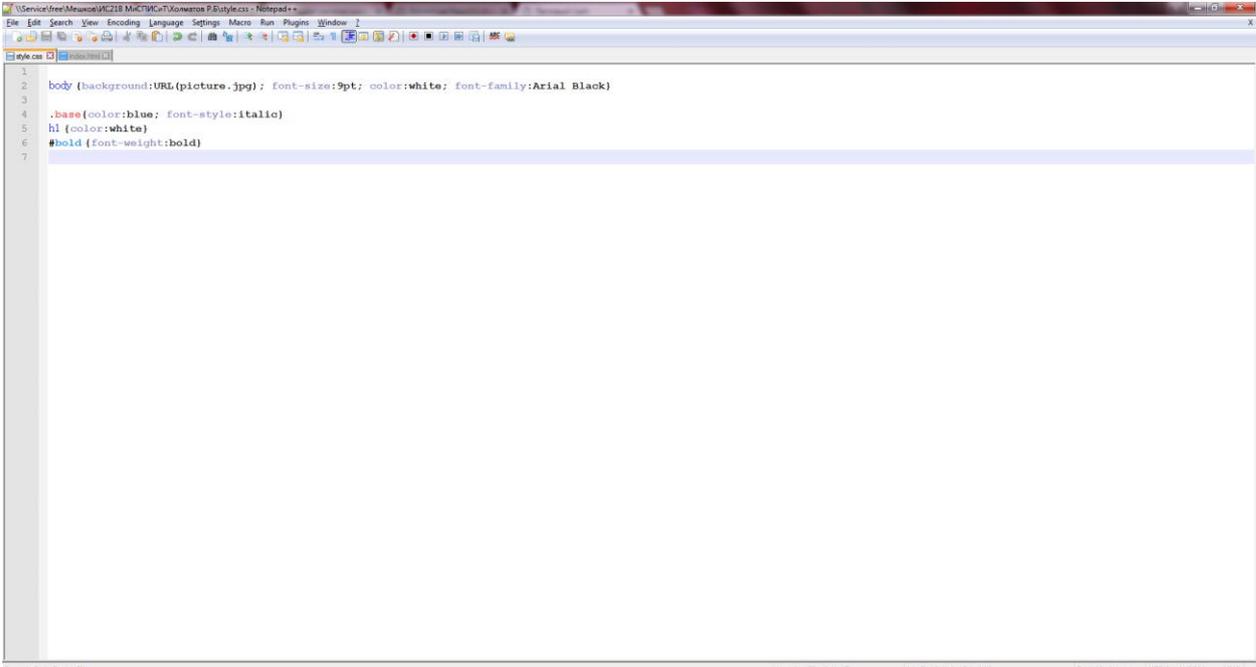
5.1 Задания.

Для разработанной в практической работе №1 интерфейсной формы создать файл style.css, в котором определить стиль для все элементов формы, рассмотренные ниже в теоретическом материале.

5.2 Теоретический материал.

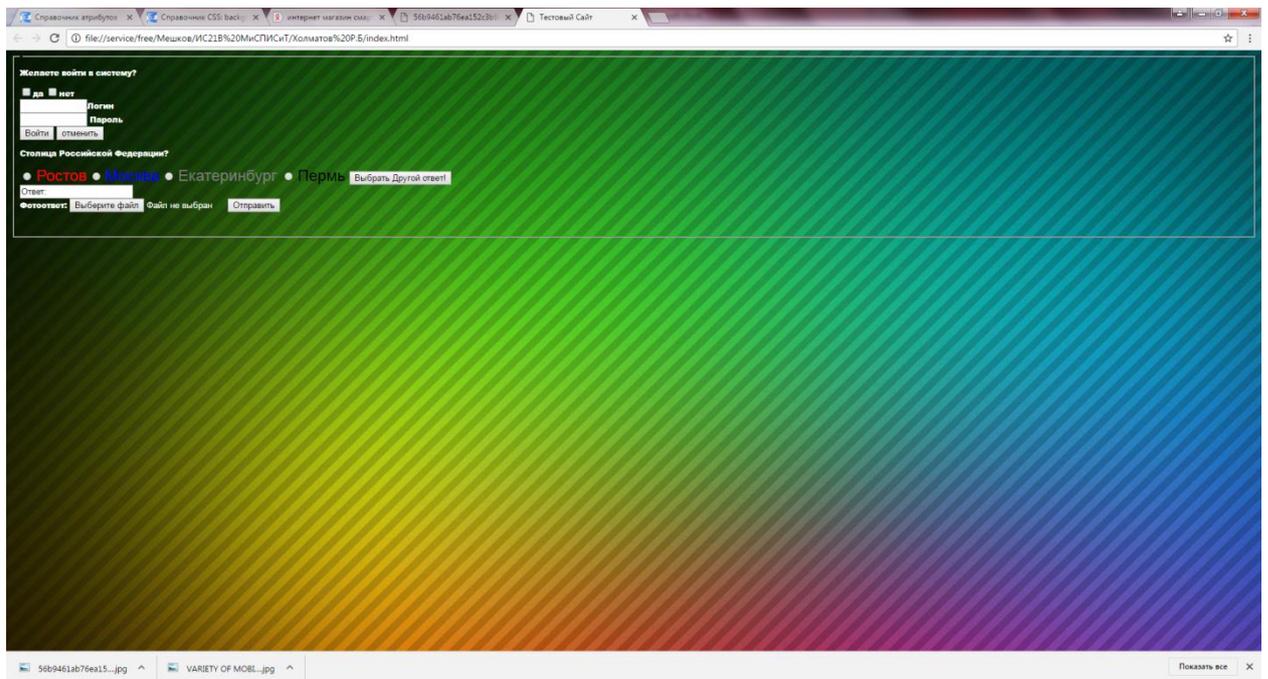
Каскадные таблицы стилей (CSS)

background - атрибут фона



```
1 body {background:URL(picture.jpg); font-size:9pt; color:white; font-family:Arial Black}
2
3
4 .base{color:blue; font-style:italic}
5 h1 {color:white}
6 #bold {font-weight:bold}
7
```

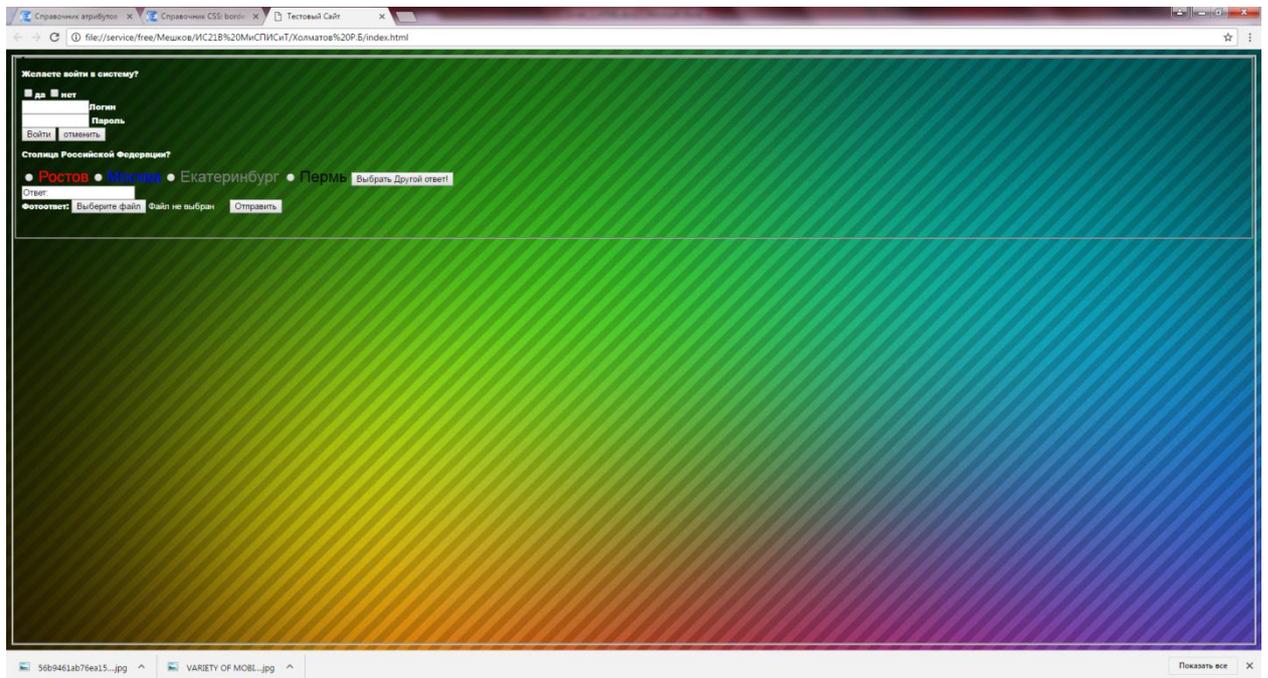
The image shows a Notepad++ window with a CSS file named 'style.css'. The code defines styles for the body, a class '.base', an element 'h1', and an ID '#bold'. The body style includes a background image, font size, color, and font family. The '.base' class sets color and font style. The 'h1' element sets color. The '#bold' ID sets font weight. The status bar at the bottom indicates 'length:175 lines:7', 'Ln:7 Col:1 Sel:0|0', 'Doc:Windows', 'UTF-8 w/o BOM', and 'ANSI'.



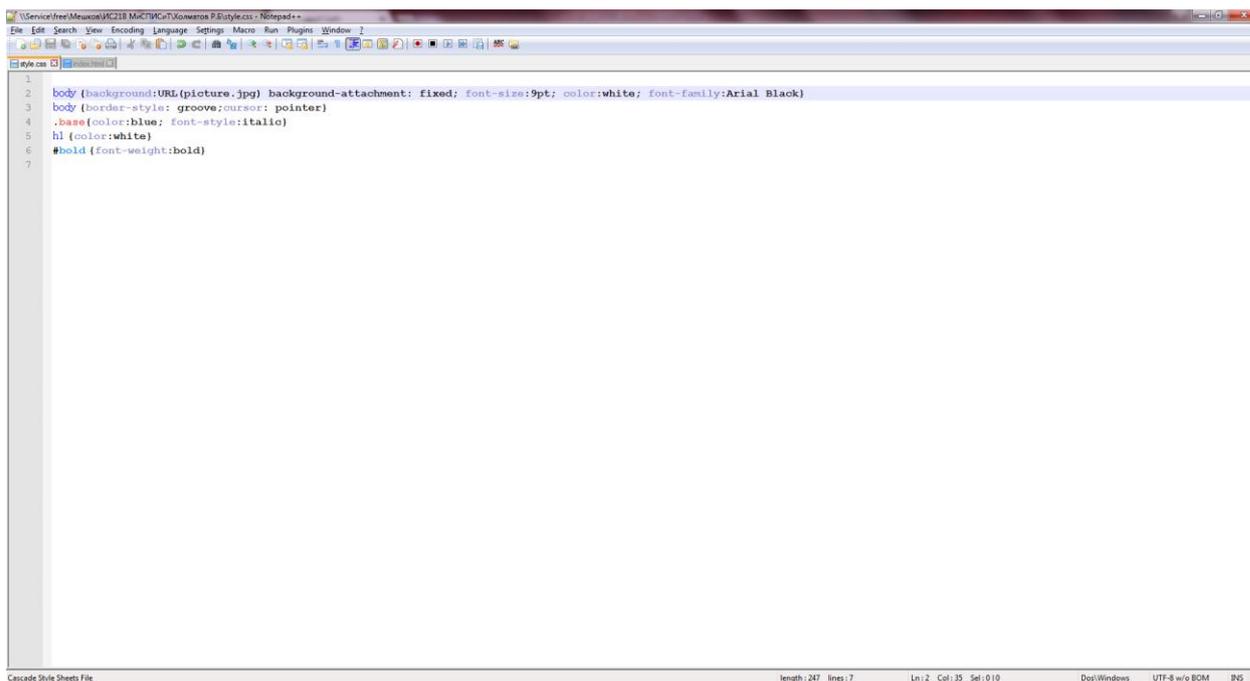
border-style - вид рамки элемента

```
\\Service\free\Мешков\ИС218\МиСПИСиТ\Холматов\Р\B\style.css - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window 2
style.css
1
2 body {background:URL(picture.jpg); font-size:9pt; color:white; font-family:Arial Black}
3 body {border-style: groove;}
4 .base {color:blue; font-style:italic}
5 h1 {color:white}
6 #bold {font-weight:bold}
7

Cascade Style Sheets File length: 203 lines: 7 Ln: 3 Col: 29 Sel: 0 | 0 Doc: Windows UTF-8 w/o BOM INS
```



background-attachment - поведение фона при прокрутке страницы



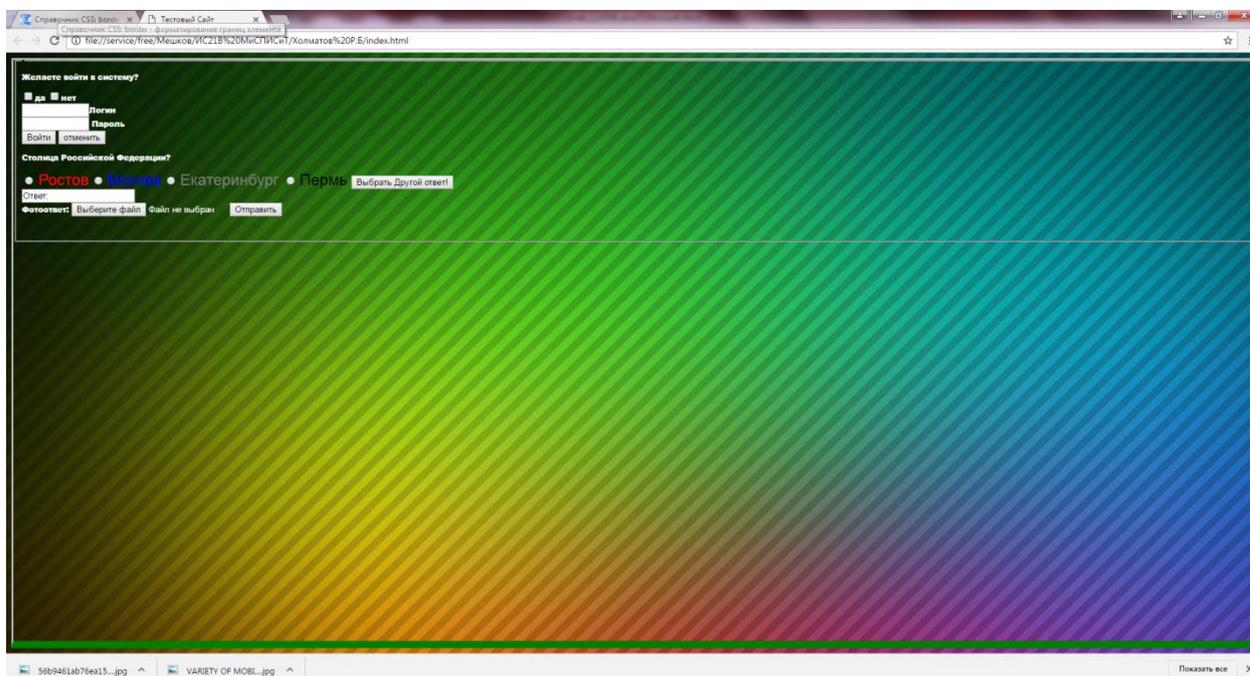
```
1 body {background:URL(picture.jpg) background-attachment: fixed; font-size:9pt; color:white; font-family:Arial Black}
2
3 body {border-style: groove;cursor: pointer}
4 .base{color:blue; font-style:italic}
5 hi {color:white}
6 #bold {font-weight:bold}
7
```

The screenshot shows a Notepad++ window with a menu bar (File, Edit, Search, View, Encoding, Language, Settings, Macro, Run, Plugins, Window) and a toolbar. The main text area contains seven lines of CSS code. The first line is highlighted in blue. The status bar at the bottom indicates 'length: 247 lines: 7', 'Ln: 2 Col: 35 Sel: 0 | 0', 'Dot/Windows', 'UTF-8 w/o BOM', and 'ANSI'.

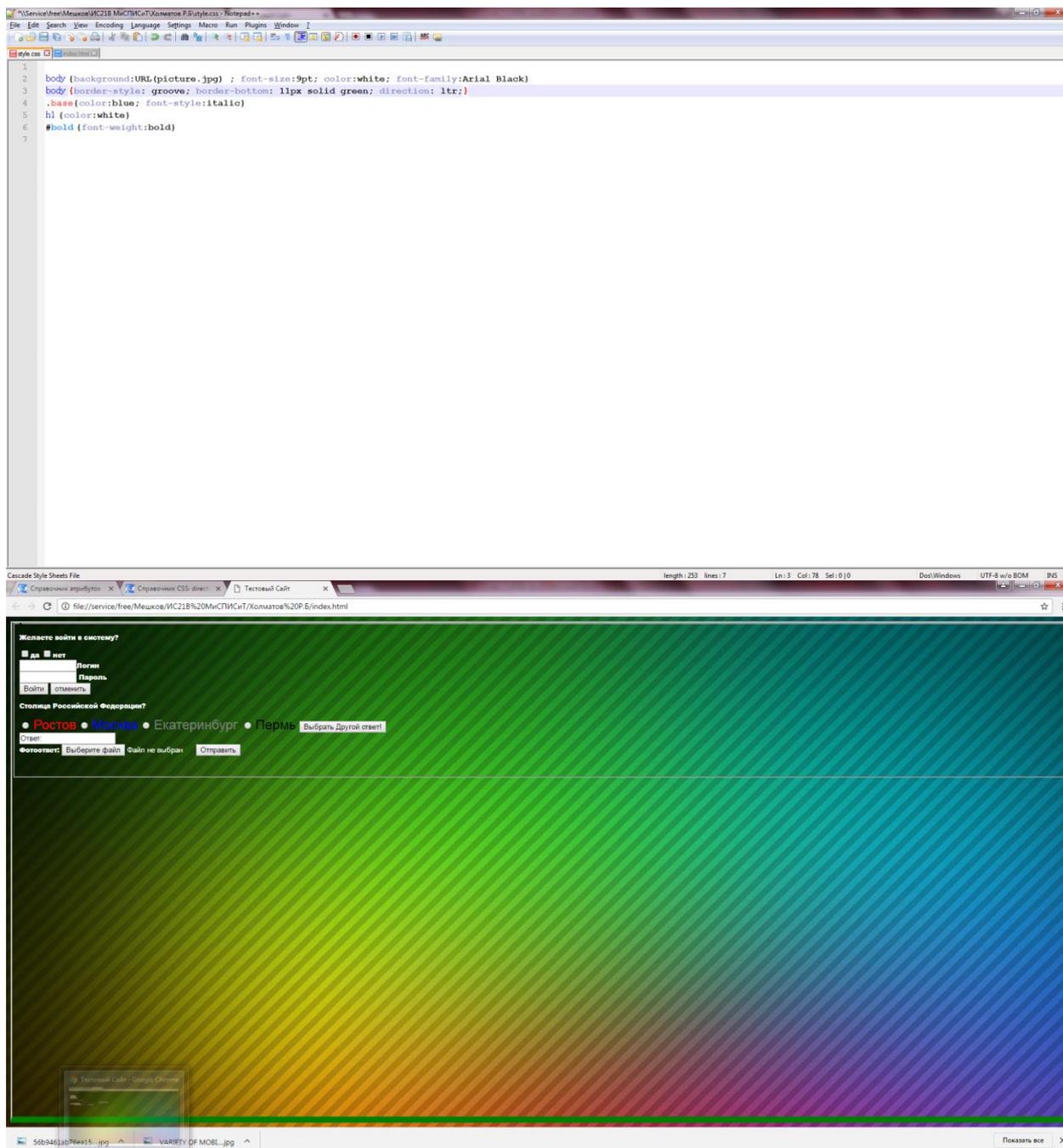
border - форматирование границ элемента

```
1
2 body {background:URL(picture.jpg) ; font-size:9pt; color:white; font-family:Arial Black}
3 body {border-style: groove; border-bottom: 11px solid green;}
4 .base {color:blue; font-style:italic}
5 h1 {color:white}
6 #bold {font-weight:bold}
7
```

Cascade Style Sheets File length: 237 lines: 7 Ln: 3 Col: 50 Sel: 0 | 0 Doc: Windows UTF-8 w/o BOM INS



direction - направление вывода текста



display - вид отображения элемента

The image shows a Notepad++ window with a CSS file named 'style.css'. The code defines styles for the body, a class named 'base', and a class named 'hold'. The body has a background image, font size of 9pt, white color, and Arial Black font family. The 'base' class has a blue color and italic font style. The 'hold' class has a bold font weight. A small preview window is visible in the bottom left of the Notepad++ window.

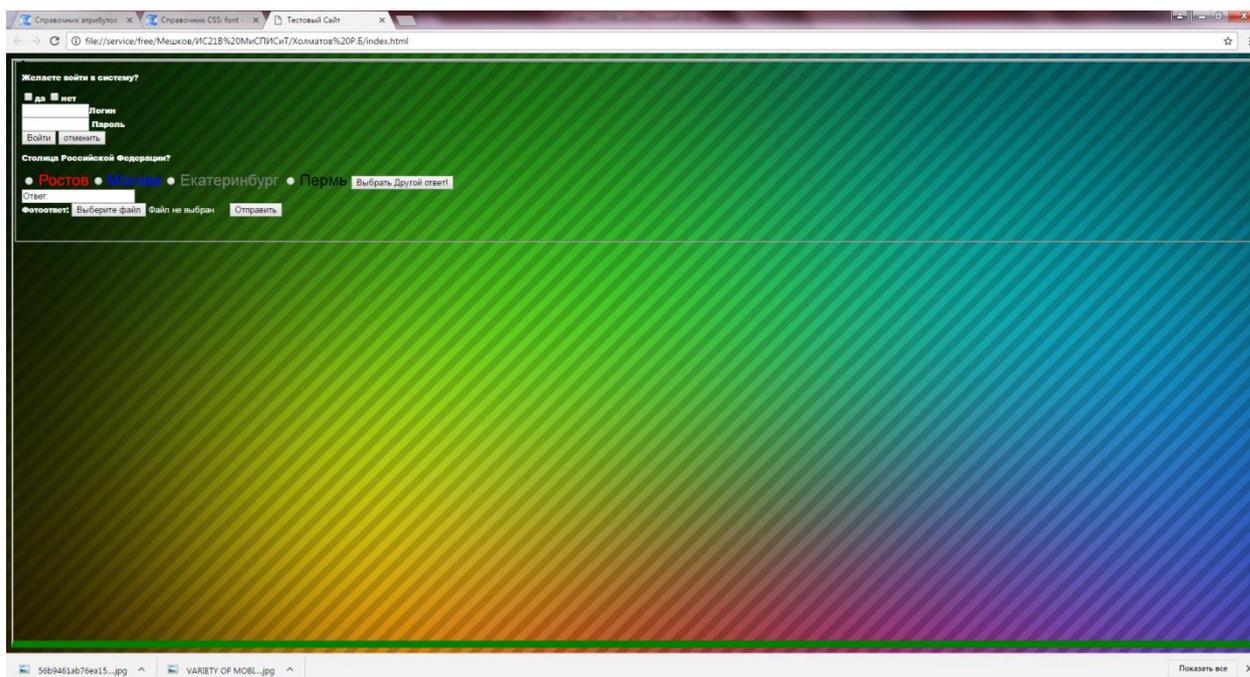
```
1  
2 body {background:URL(picture.jpg) ; font-size:9pt; color:white; font-family:Arial Black}  
3 body {border-style: groove; border-bottom: 11px solid green; direction: ltr;}  
4 body {display: block}  
5 .base {color:blue; font-style:italic}  
6 h1 {color:white}  
7 #hold {font-weight:bold}  
8
```

The browser window below shows the rendered page. The background is a green-to-blue gradient with diagonal lines. The page content includes a login form with fields for 'логин' and 'пароль', a 'Войти' button, and a 'Страна' dropdown menu with options for 'Ростов', 'Москва', 'Екатеринбург', and 'Пермь'. There is also a 'Файл' section with a 'Выберите файл' button and an 'Отправить' button.

font - форматирование шрифта

```
1 body {background:URL(picture.jpg)}
2
3 body {font: Tahoma, sans-serif italic 10px; ; font-size:9pt; color:white; font-family:Arial Black}
4 body {border-style: groove; border-bottom: 11px solid green; direction: ltr;}
5 body {display: block}
6 .base{color:blue; font-style:italic}
7 h1 {color:white}
8 #bold {font-weight:bold}
9
```

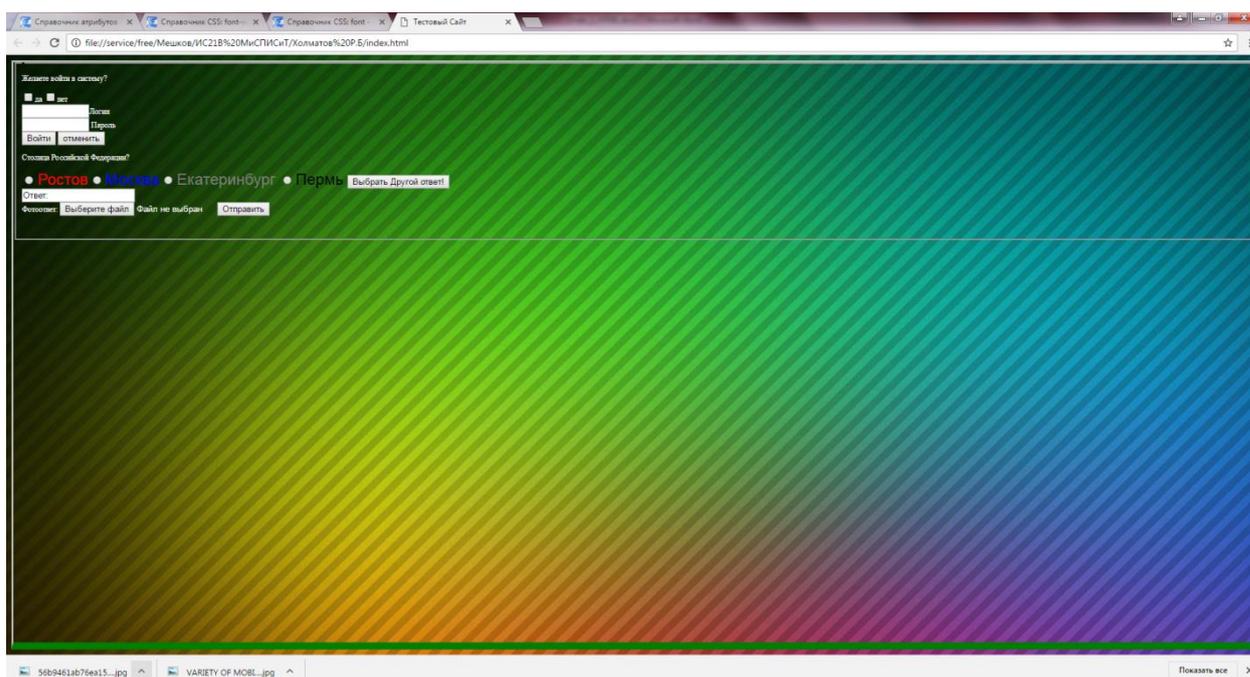
Cascade Style Sheets File length: 321 lines: 9 Ln:3 Col:19 Sel:0|0 Doc:Windows UTF-8 w/o BOM IN5



font-weight - насыщенность шрифта

```
1
2 body {background:URL(picture.jpg) !
3 body {font: Tahoma,sans-serif italic 10px; ; font-size:9pt; color:white; font-family:Arial Black font-weight: 400}
4 body {border-style: groove; border-bottom: 11px solid green; direction: ltr;}
5 body {display: block}
6 .base{color:blue; font-style:italic}
7 h1 {color:white}
8 #bold {font-weight:bold}
9
```

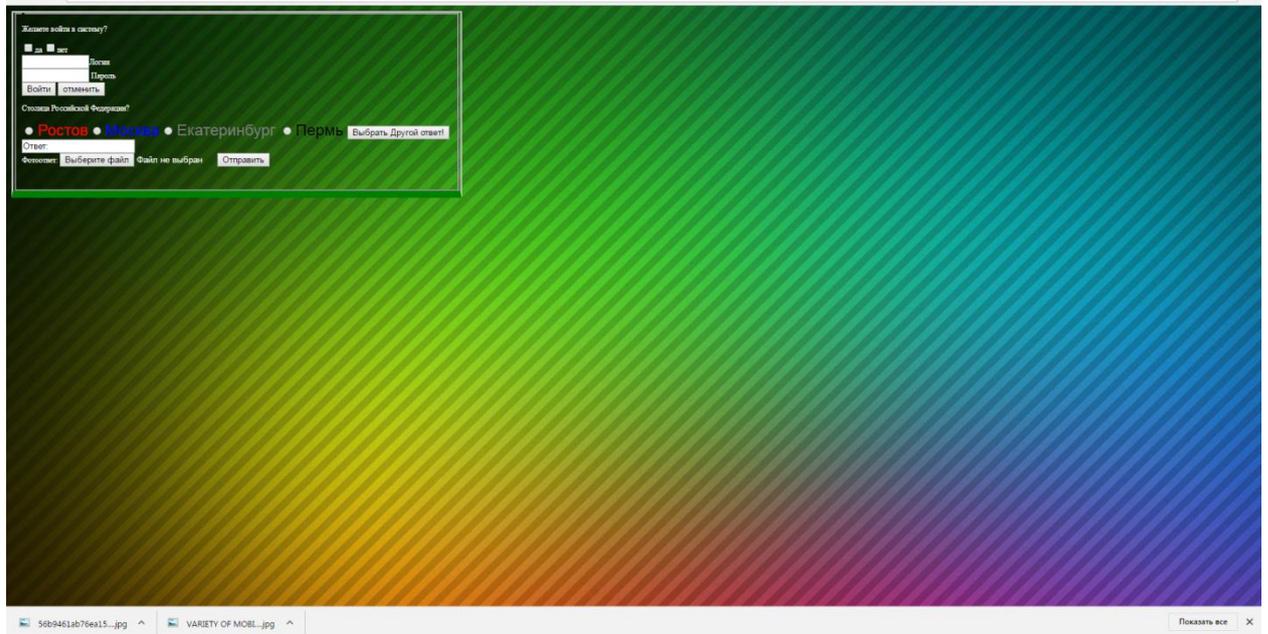
Cascade Style Sheets File length: 338 lines: 9 Ln: 3 Col: 112 Sel: 0 | 0 Dos/Windows UTF-8 w/o BOM INS



left - расстояние от левого края

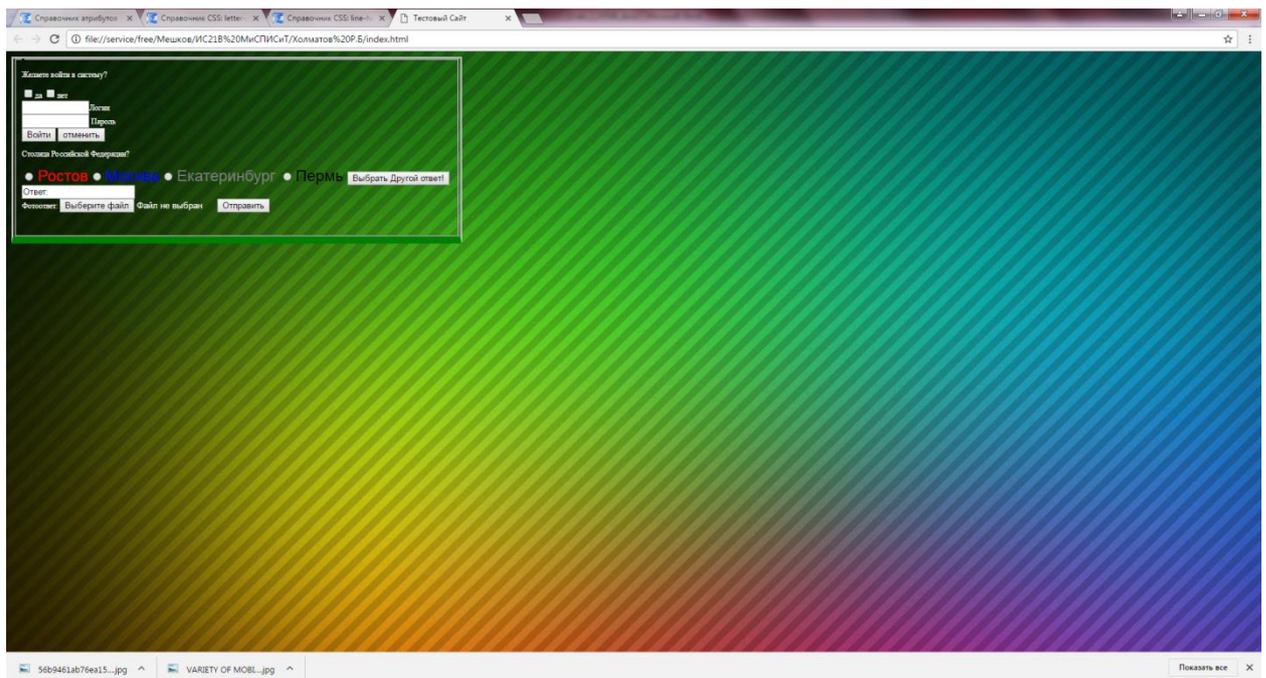
```
1
2 body {background:URL(picture.jpg)}
3 body {font: Tahoma,sans-serif italic 10px; ; font-size:9pt; color:white; font-family:Arial Black font-weight: 400}
4
5 body {border-style: groove; border-bottom: 11px solid green; direction: ltr;}
6 body {display: block; position: absolute; left: 10 px}
7 .base{color:blue; font-style:italic}
8 hi {color:white}
9 #bold {font-weight:bold}
10
```

Cascade Style Sheets File length: 373 lines: 10 Ln: 6 Col: 34 Sel: 0 | 0 Dot: Windows UTF-8 w/o BOM INS



letter-spacing - межсимвольный интервал

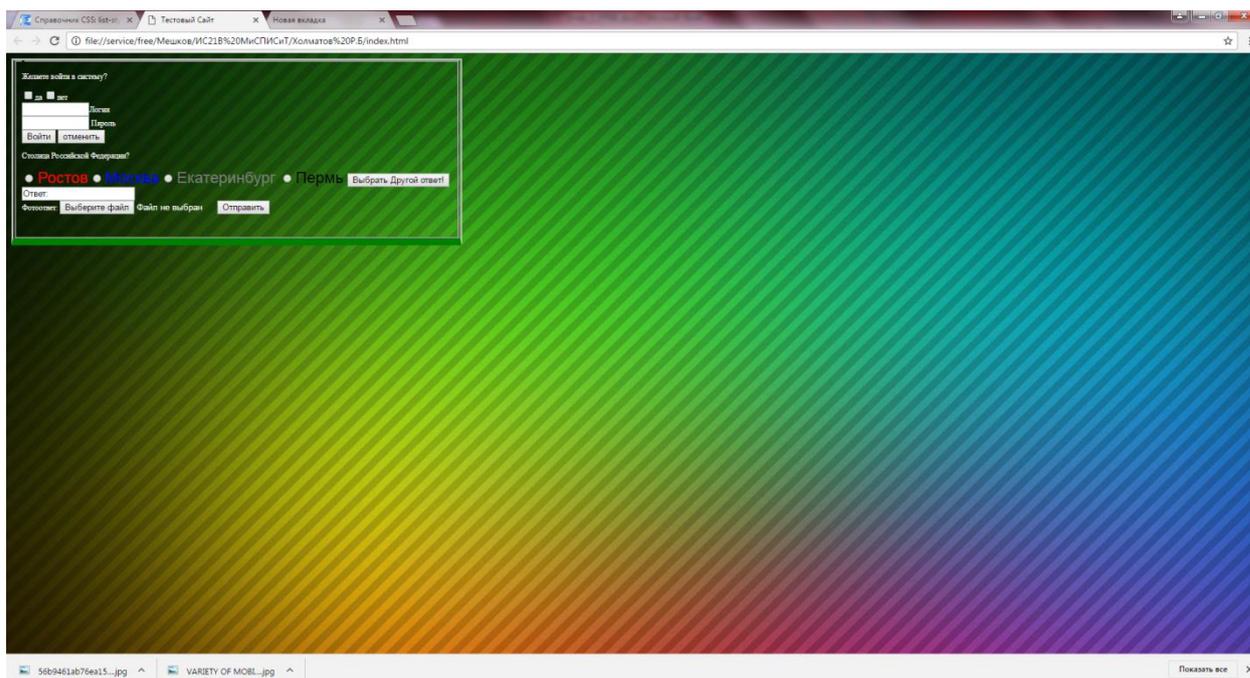
```
1 \\Service\free\Мешков\ИС218\МисПСИСиТ\Холматов\Р.Б\style.css - Notepad++
2
3 body {background:URL(picture.jpg)}
4
5 body {font: Tahoma,sans-serif italic 10px; ; font-size:9pt; color:white; font-family:Arial Black font-weight: 400; letter-spacing: 5 px}
6
7 body {border-style: groove; border-bottom: 11px solid green; direction: ltr;}
8
9 body {display: block; position: absolute; left: 10 px}
10 .base{color:blue; font-style:italic}
11
12 h1 {color:white}
13
14 #hold {Font-weight:bold}
15
```



list-style - вид списочного маркера

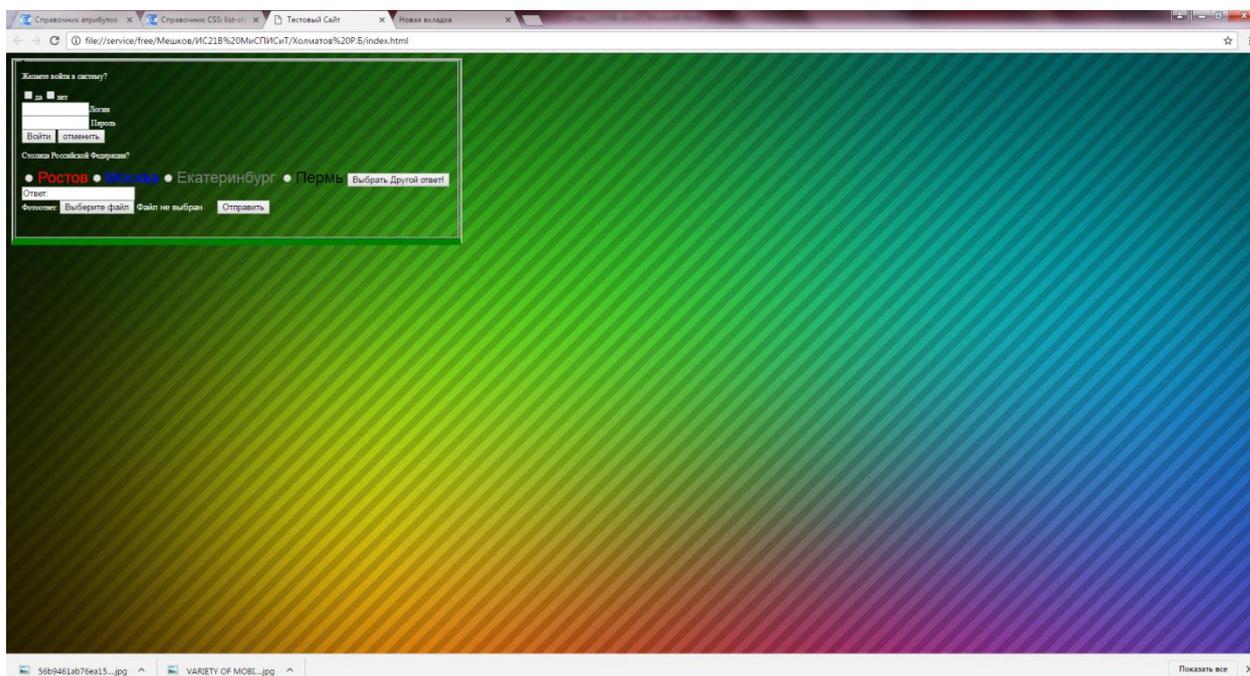
```
1
2 body {background:URL(picture.jpg)}
3 body {font: Tahoma,sans-serif italic 10px; ; font-size:9pt; color:white; font-family:Arial Black font-weight: 400; letter-spacing: 5 px; list-style: outside decimal}
4
5 body {border-style: groove; border-bottom: 11px solid green; direction: ltr;}
6 body {display: block; position: absolute; left: 10 px}
7 .base{color:blue; font-style:italic}
8 h1 {color:white}
9 #bold {font-weight:bold}
10
```

Cascade Style Sheets File length: 424 lines: 10 Ln:3 Col:165 Sel:0|0 Dot/Windows UTF-8 w/o BOM IN5



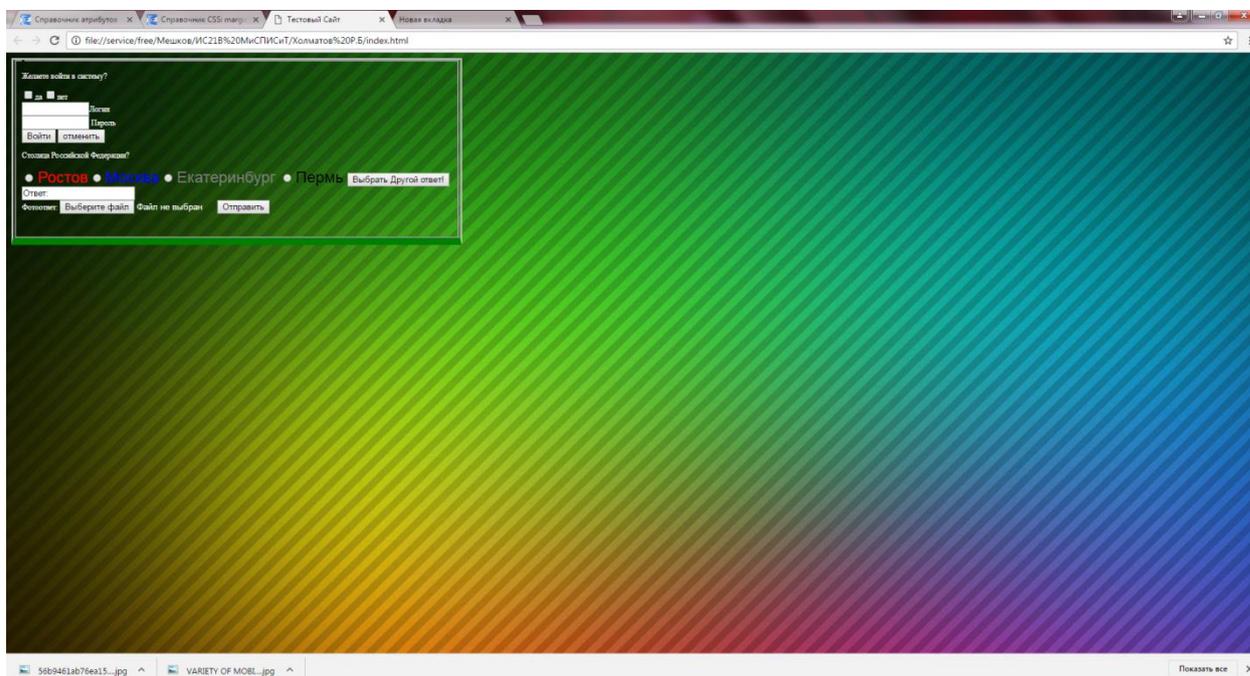
list-style-position - положение маркера в списке

```
1
2 body {background:URL(picture.jpg)}
3 body {font: Tahoma,sans-serif italic 10px; ; font-size:9pt; color:white; font-family:Arial Black font-weight: 400; letter-spacing: 5 px; list-style: outside decimal}
4
5 body {border-style: groove; border-bottom: 11px solid green; direction: ltr;}
6 body {display: block; position: absolute; left: 10 px; list-style-position: inside}
7 .base{color:blue; font-style:italic}
8 h1 {color:white}
9 #bold {font-weight:bold}
10
```



margin - внешний отступ

```
1 body {background:URL(picture.jpg)}
2
3 body {font: Tahoma,sans-serif italic 10px; ; font-size:9pt; color:white; font-family:Arial Black font-weight: 400; letter-spacing: 5 px; list-style: outside decimal}
4 body {margin: 15 pt 0}
5
6 body {border-style: groove; border-bottom: 11px solid green; direction: ltr;}
7
8 #base {color:blue; font-style:italic;
9
10 #bold {font-weight:bold}
```



overflow - управление выводом содержания блочного элемента

The image shows a Notepad++ window with the following CSS code:

```
1  
2 body {background:URL(picture.jpg)}  
3 body {font: Tahoma,sans-serif italic 10px; ; font-size:9pt; color:white; font-family:Arial Black font-weight: 400; letter-spacing: 5 px; list-style: outside decimal}  
4 body {margin: 15 pt 0; overflow: hidden}  
5 body {border-style: groove; border-bottom: 11px solid green; direction: ltr;}  
6 body {display: block; position: absolute; left: 10 px; list-style-position: inside}  
7 .base{color:blue; font-style:italic}  
8 h1 {color:white}  
9 #hold {font-weight:bold}  
10
```

Below the code editor, a browser window displays the rendered page. The page features a background image of a rainbow gradient with diagonal lines. A small white box in the top-left corner contains a login form with the following elements:

- Buttons: Войти, Отмена
- Text: Хотите войти в систему?
- Form fields: Имя, Пароль
- Text: Выберите файл
- Text: Выберите файл
- Text: Отправить

text-decoration - вид оформления текста

The image shows a Notepad++ window with the following CSS code:

```
1 body {background:URL(picture.jpg)}
2
3 body {font: Tahoma,sans-serif italic 10px; ; font-size:9pt; color:white; font-family:Arial Black font-weight: 400; letter-spacing: 5 px; list-style: outside decimal}
4 body {margin: 15 pt 0; overflow: hidden; text-decoration: underline}
5 body {border-style: groove; border-bottom: 11px solid green; direction: ltr;}
6 body {display: block; position: absolute; left: 10 px; list-style-position: inside}
7 .base{color:blue; font-style:italic}
8 h1 {color:white}
9 #bold {font-weight:bold}
10
```

Below the Notepad++ window is a browser window displaying a login form on a background with a green-to-blue gradient. The form includes:

- Fields for "Имя" (Name) and "Пароль" (Password).
- Buttons for "Войти" (Login) and "Отменить" (Cancel).
- A section for "Список российских городов" (List of Russian cities) with radio buttons for "Ростов", "Москва", "Екатеринбург", and "Пермь", and a "Выбрать Другой ответ" (Select other answer) option.
- Fields for "Страна" (Country) and "Файл" (File) with a "Выберите файл" (Select file) button and an "Отправить" (Submit) button.

Практическая работа № 6. Передача параметров от HTML формы к PHP процессу.

Цель: изучить взаимодействие программ в много файловом Web проекте.

6.1 Задания.

Для заданной вариантом предметной области и базе данных, построенной в практической работе №2 модифицировать интерфейсную форму (практическая работа №1) и написать на языке PHP программу доступа к SQL базе данных. Использовать модификацию файла style.css из практической работы №5 для управления стилем формы. Вариант определяется в приведенной ниже таблице по последнему номеру зачетной книжки студента.

Последняя цифра в зачетной книжке студента	Номер варианта	Предметная область
1	1	Мониторы
2	2	Принтеры
3	3	Планшеты
4	4	Ноутбуки
5	5	Манипуляторы «мышь»
6	6	Центральные процессорные блоки
7	7	Смартфоны
8	8	Коммутаторы (хабы)
9	9	Клавиатуры
0	10	Сканеры

6.2 Теоретический материал.

Создание много файлового проекта

Для разбиения на отдельные файлы необходимо выполнить следующие операции.

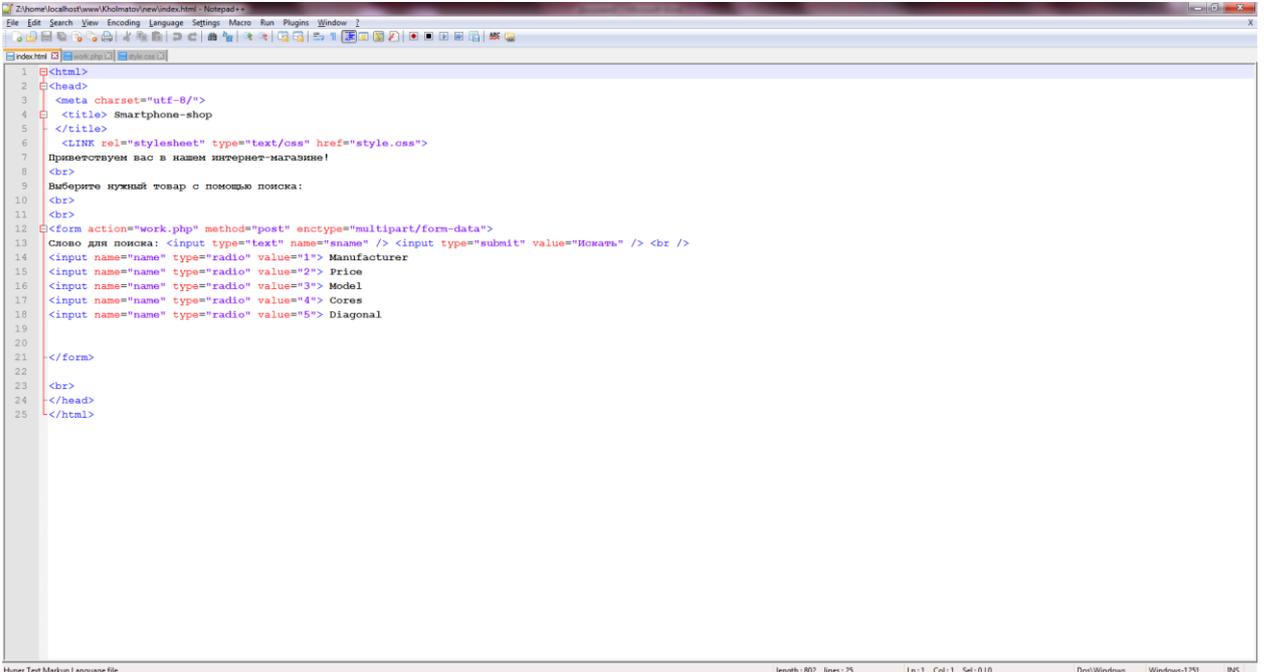
1. Создаем новый файл «index.html»

В HEAD указываем путь к файлу «.css»

<LINK rel="stylesheet" type="text/css" href="style.css">

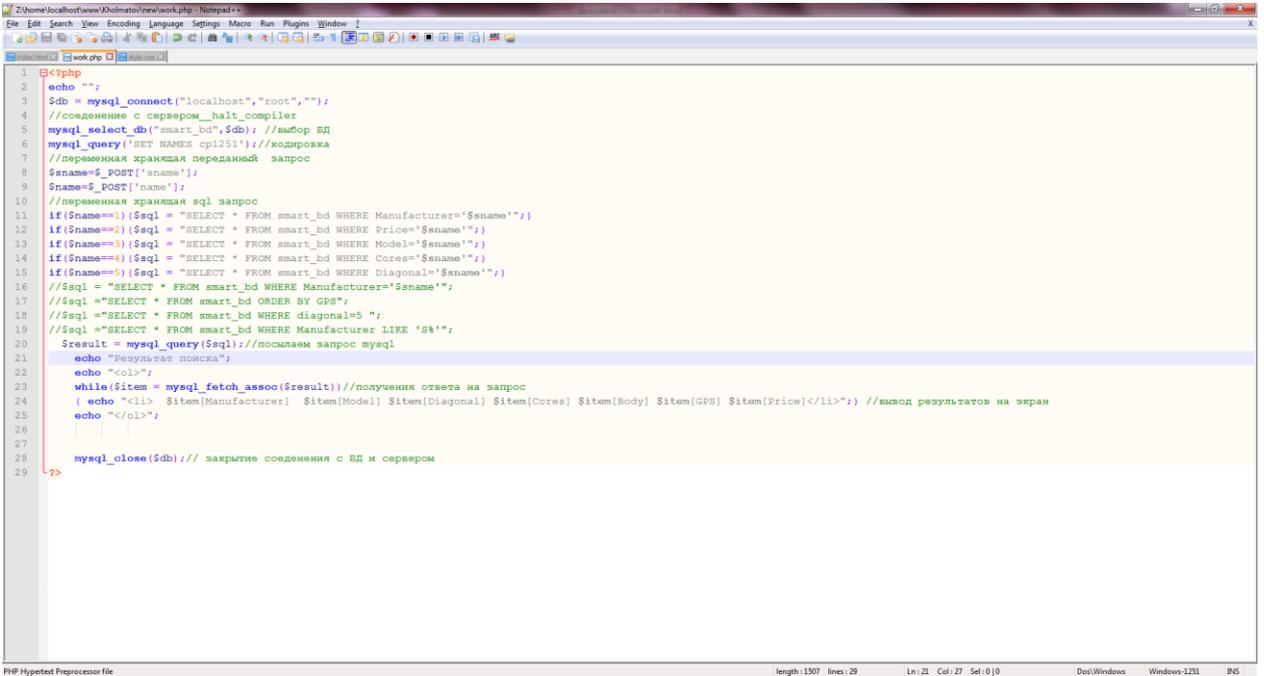
В action указываем путь к файлу «.php»

<form action="work.php" method="post" enctype="multipart/form-data">



```
1 <html>
2 <head>
3   <meta charset="utf-8/">
4   <title> Smartphone-shop
5 </title>
6   <LINK rel="stylesheet" type="text/css" href="style.css">
7   Приветствуем вас в нашем интернет-магазине!
8   <br>
9   Выберите нужный товар с помощью поиска:
10  <br>
11  <br>
12  <form action="work.php" method="post" enctype="multipart/form-data">
13  Словно для поиска: <input type="text" name="sname" /> <input type="submit" value="Искать" /> <br />
14  <input name="name" type="radio" value="1"> Manufacturer
15  <input name="name" type="radio" value="2"> Price
16  <input name="name" type="radio" value="3"> Model
17  <input name="name" type="radio" value="4"> Cores
18  <input name="name" type="radio" value="5"> Diagonal
19
20
21 </form>
22
23 <br>
24 </head>
25 </html>
```

2. Создаем файл «work.php»

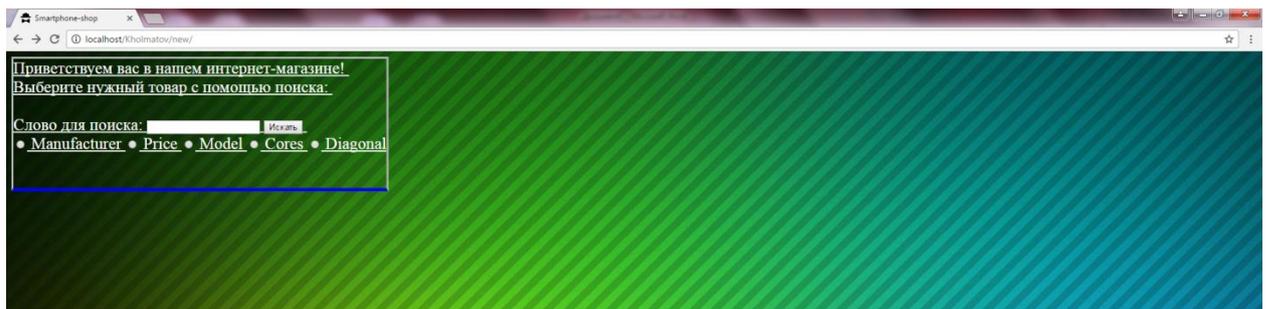


```
1 <?php
2 echo "";
3 $db = mysql_connect("localhost","root","");
4 //соединение с сервером_half_compiler
5 mysql_select_db("smart_bd",$db); //выбор БД
6 mysql_query('SET NAMES cp1251');//кодировка
7 //переменная хранящая переданный запрос
8 $sname=$_POST['sname'];
9 $sname=$_POST['name'];
10 //переменная хранящая sql запрос
11 if($sname==1){$sql = "SELECT * FROM smart_bd WHERE Manufacturer='$sname'";}
12 if($sname==2){$sql = "SELECT * FROM smart_bd WHERE Price='$sname'";}
13 if($sname==3){$sql = "SELECT * FROM smart_bd WHERE Model='$sname'";}
14 if($sname==4){$sql = "SELECT * FROM smart_bd WHERE Cores='$sname'";}
15 if($sname==5){$sql = "SELECT * FROM smart_bd WHERE Diagonal='$sname'";}
16 // $sql = "SELECT * FROM smart_bd WHERE Manufacturer='$sname'";
17 // $sql = "SELECT * FROM smart_bd ORDER BY GPS";
18 // $sql = "SELECT * FROM smart_bd WHERE diagonal=5 ";
19 // $sql = "SELECT * FROM smart_bd WHERE Manufacturer LIKE 's%'";
20 $result = mysql_query($sql);//посылаем запрос mysql
21 echo "Результат поиска";
22 echo "<br>";
23 while($item = mysql_fetch_assoc($result)){//получения ответа на запрос
24   { echo "<li> $item[Manufacturer] $item[Model] $item[Diagonal] $item[Cores] $item[GPS] $item[Price]</li>"; //вывод результатов на экран
25   echo "<br>";
26   }
27 }
28 mysql_close($db);// закрытие соединения с БД и сервером
29 >>
```

3. Создаём файл «style.css»

```
1 body {background: URL(picture.jpg)}
2 body {font: Tahoma,sans-serif italic 20px; ; font-size:19pt; color:white; font-family:Arial Black font-weight: 400; letter-spacing: 5 px; list-style: outside decimal}
3 body {margin: 15 pt 0; overflow: hidden; text-decoration: underline}
4 body {border-style: groove; border-bottom: 5px solid blue; direction: ltr;}
5 body {display: block; position: absolute; left: 10 px; list-style-position: inside}
6 .base {color:blue; font-style:italic}
7 h1 {color:white}
8 #bold {font-weight:bold}
9
```

Получаем результат, как мы видим html работает.



Взаимодействие с базой данных осуществляется успешно.



Практическая работа № 7. Программное формирование запросов к базе данных SQL из программы на языке PHP.

Цель: изучить программное формирование запросов к базе данных SQL с помощью функционала языка PHP.

7.1 Задания.

Для заданной вариантом предметной области и базе данных, построенной в практической работе №2 использовать интерфейсную форму (практическая работа №6) и написать на языке PHP программу формирования запросов к SQL базе данных. Использовать модификацию файла style.css из практической работы №5 для управления стилем формы. Вариант определяется в приведенной ниже таблице по последнему номеру зачетной книжки студента.

Последняя цифра в зачетной книжке студента	Номер варианта	Предметная область
1	1	Мониторы
2	2	Принтеры
3	3	Планшеты
4	4	Ноутбуки
5	5	Манипуляторы «мышь»
6	6	Центральные процессорные блоки
7	7	Смартфоны
8	8	Коммутаторы (хабы)
9	9	Клавиатуры
0	10	Сканеры

7.2 Теоритический материал.

Взаимодействие с базой данных SQL

Нам нужно создать новый интерфейс, соответствующий данной работе

Создаём файл index.html Заполняем

```
1 <html>
2 <head>
3 <LINK rel="stylesheet" type="text/css" href="style.css">
4 </head>
5 <body>
6
7 <br>
8 <br>
9 <form action="action.php" method="post" enctype="multipart/form-data">
10 Цена: <input type="text" name="Price" /><br />
11 Производитель: <input type="text" name="Manufacturer" /><br />
12 Модель: <input type="text" name="Model" /><br />
13 Диагональ: <input type="text" name="Diagonal" /><br />
14 Ядро: <input type="text" name="Cores" /><br />
15 Корпус: <input type="text" name="Body" /><br />
16 GPS: <input type="text" name="GPS" /><br />
17 <input type="submit" value="ДОБАВИТЬ" />
18 </form>
19
20 <br>
21 </body>
22 </html>
23
```

Создаём файл action.php в котором указываем базу данных и таблицу в которую будем добавлять значения.

Заполняем

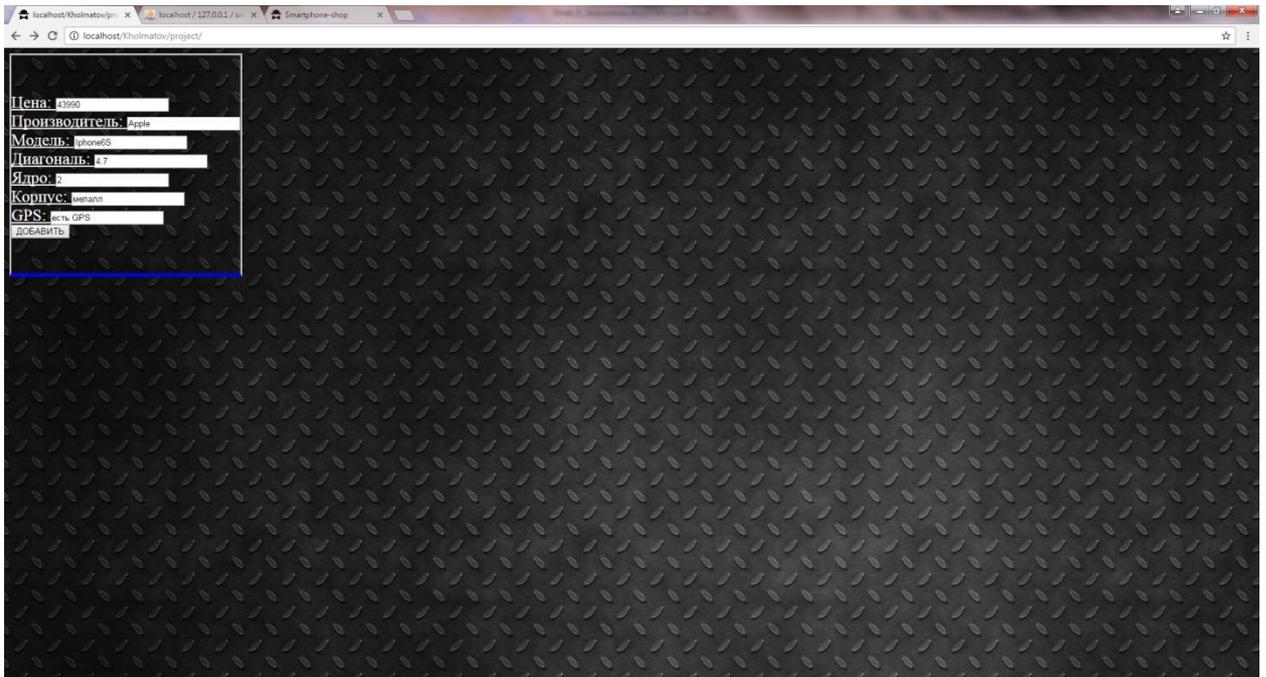
```
1 <?php
2
3 echo "<br>";
4 $db = mysql_connect("localhost","root","");
5 //соединение с сервером_half_compiler
6 mysql_select_db("smart_bd",$db); //выбор БД
7 mysql_query('SET NAMES cp1251');//кодировка
8 //переменная хранящая переданный запрос
9
10 $Price=$_POST['Price'];
11 $Manufacturer=$_POST['Manufacturer'];
12 $Model=$_POST['Model'];
13 $Diagonal=$_POST['Diagonal'];
14 $Cores=$_POST['Cores'];
15 $Body=$_POST['Body'];
16 $GPS=$_POST['GPS'];
17
18 //Добавить строку в таблицу
19
20 $sql="INSERT INTO smart_bd VALUES ('$Price', '$Manufacturer', '$Model', '$Diagonal', '$Cores', '$Body', '$GPS')";
21 mysql_query($sql); //Строка добавлена
22 //переменная хранящая sql запрос
23 // $sql = "SELECT Price, Manufacturer, Model, Diagonal, Cores, Body, GPS FROM smart_bd";
24 $sql = "SELECT * FROM smart_bd";
25 $result = mysql_query($sql);//посылаем запрос mysql
26 echo "Результат:";
27 echo "<br>";
28 while($item = mysql_fetch_assoc($result)){//получения элемента выборки
29     echo "<li>$item[Price], $item[Manufacturer], $item[Model], $item[Diagonal], $item[Cores], $item[Body], $item[GPS]</li>"; //вывод результатов на экран
30 }
31 echo "<br>";
32 mysql_close($db); // закрываем соединения с БД и сервером
33
```

Добавляем стиль Style.css

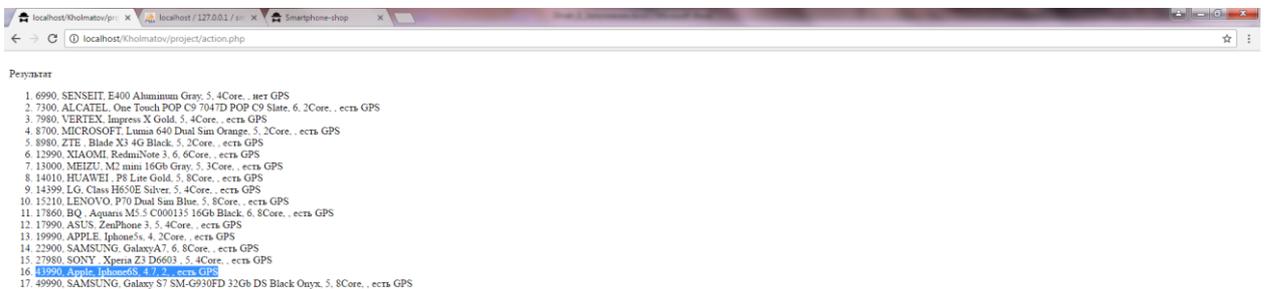
```
1 body {background:URL(picture.jpg)}
2 body {font: Tahoma,sans-serif italic 20px; ; font-size:19pt; color:white; font-family:Arial Black font-weight: 400; letter-spacing: 5 px; list-style: outside decimal}
3 body {margin: 15 pt 0; overflow: hidden; text-decoration: underline}
4 body {border-style: groove; border-bottom: 5px solid blue; direction: ltr;}
5 body {display: block; position: absolute; left: 10 px; list-style-position: inside}
6 .base{color:blue; font-style:italic}
7 h1 {color:white}
8 #bold {font-weight:bold}
9
```

Получаем вот такой интерфейс:

Добавляем в Smart_bd



Результат:



localhost/127.0.0.1/... smart_bd | SmartPhone-shop

phpMyAdmin

smart_bd

Обзор Структура SQL Поиск Вставить Экспорт Импорт Операции Слежение Траггеры

Отображает строки 0 - 16 (-17 всего) | Запрос занял 0.0005 сек. | [Manufacturer: ALCATEL - ZTE]

```

SELECT *
FROM smart_bd
ORDER BY smart_bd `Manufacturer` ASC
LIMIT 0, 30

```

Показать: Начальная строка 0 Количество строк 30 Заголовки каждые 100 строк

Сортировать по индексу: Нет

Параметры

	Price	Manufacturer	Model	Diagonal	Cores	Body	GPS
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	7300	ALCATEL	One Touch POP C9 7047D POP C9 Slate	6	2Core	пластик	есть GPS
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	19990	APPLE	iPhone5s	4	2Core	металл	есть GPS
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	43990	Apple	iPhone5S	4.7	2	металл	есть GPS
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	17990	ASUS	ZenPhone 3	5	4Core	Пластик	есть GPS
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	17900	BQ	Aquaris M5 5 C000135 16Gb Black	6	8Core	металл	есть GPS
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	14010	HUAWEI	P8 Lite Gold	5	8Core	металл/пластик	есть GPS
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	15210	LENOVO	P70 Dual Sim Blue	5	8Core	пластик	есть GPS
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	14399	LG	Class H650E Silver	5	4Core	пластик	есть GPS
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	13000	MEIZU	M2 mini 16Gb Gray	5	3Core	металл	есть GPS
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	8700	MICROSOFT	Lumia 640 Dual Sim Orange	5	2Core	пластик	есть GPS
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	49990	SAMSUNG	Galaxy S7 SM-G930FD 32Gb DS Black Onyx	5	8Core	металл/пластик	есть GPS
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	22900	SAMSUNG	GalaxyA7	6	8Core	металл	есть GPS
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	6990	SENSEIT	E400 Aluminum Gray	5	4Core	пластик	нет GPS
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	27900	SONY	Xperia Z3 D6603	5	4Core	металл	есть GPS
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	7900	VERTEX	Impress X Gold	5	4Core	пластик	есть GPS
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	12900	XIACMI	RedmiNote 3	6	6Core	Пластик/металл	есть GPS
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	8900	ZTE	Blade X3 4G Black	5	2Core	пластик	есть GPS

Отметить все / Снять выделение / С отмененных: Изменить Удалить Экспорт

Показать: Начальная строка 0 Количество строк 30 Заголовки каждые 100 строк

Использование результатов запроса

Версия для печати | Версия для печати (полностью) | Экспорт | Отобразить график | Создать представление

ЛИТЕРАТУРА

Основная литература

- 1 Золотов, С.Ю. Проектирование информационных систем Томск : «Эль Контент» учебное пособие 2013
<http://biblioclub.ru/index.php?page=book&id=208706> [Электронный ресурс]
- 2 Стасышин, В.М. Проектирование информационных систем и баз данных Новосибирск : «НГТУ» учебное пособие 2012
<http://biblioclub.ru/index.php?page=book&id=228774> [Электронный ресурс]

Дополнительная литература

- 3 Абдикеев, Н. М. Проектирование интеллектуальных систем в экономике М.: «Экзамен» учебник для студентов вузов 2004 [Текст]
- 4 Банк, В.Р. Информационные системы в экономике М.: «Экономистъ» учебник для студентов вузов 2005 [Текст]
- 5 Бенкен, Е. С. PHP, MySQL, XML: программирование для Интернета СПб. : «БХВ-Петербург» 2011 [Текст]
- 6 Блюмин, А. М. Проектирование систем информационного, консультационного и инновационного обслуживания М.: «Дашков и К» учеб. пособ. 2010 [Текст]
- 7 олодурина, И. Проектирование компонентов распределенных информационных систем Оренбург : ОГУ учеб. пособие 2012
<http://biblioclub.ru/index.php?page=book&id=259156> [Электронный ресурс]
- 8 Бурков А.В. Проектирование информационных систем в Microsoft SQL Server 2008 и Visual Studio 2008 М.: Интернет-Университет Информационных Технологий (ИНТУИТ) 2016
<http://www.iprbookshop.ru/16730> [Электронный ресурс]
- 9 Вендров, А.М Проектирование программного обеспечения

экономических информационных систем М.: «Финансы и статистика»
учебник для вузов 2002

10 Гвоздева Т. В. Проектирование информационных систем Ростов
н/Д.: «Феникс» учеб. Пособие 2009

11 Грекул В.И. Проектирование информационных систем М.:
Интернет-Университет Информационных Технологий (ИНТУИТ) 2008
<http://www.iprbookshop.ru/22438> [Электронный ресурс]

12 Грекул В.И. Управление внедрением информационных
систем, системМ. : Интернет-Университет Информационных Технологий:
БИНОМ. Лаборатория знаний: учебник для вузов 2010 10 [Текст]

13 Кузнецов, М. В Объектно-ориентированное программирование на
PHP СПб. : БХВ-Петербург 2007 [Текст]

14 Маклаков, С. В Создание информационных систем с AllFusion
Modeling Suite М. : Диалог – МИФИ 2007 15 [Текст]

15 Соловьев, И.В. Проектирование информационных систем.
Фундаментальный курс. для высшей школы М. : Академический проект
учебное пособие 2009 <http://biblioclub.ru/index.php?page=book&id=144206>
[Электронный ресурс]

16 Эванс, Э. Предметно-ориентированное проектирование:
структуризация сложных программных систем М. : Вильямс 2011 2 [Текст]

Периодические издания

17 Информационные технологии и вычислительные системы 2015

18 Научно-техническая информация (НТИ). Серия 2.
Информационные процессы и системы 2015.