Министерство образования Республики Беларусь

Учреждение образования «Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники»

Факультет компьютерных технологий

Кафедра информационных систем и технологий

Дисциплина: Веб-технологии

Контрольная работа

Студент: гр. 681072 Иванов К. Е.

Проверила: Коренская И.Н.

Минск 2019

**Задание 1. Основы вёрстки.**

**Цель:** Разработать HTML-страницу, соответствующую скриншоту.

**Задание:** Разработать HTML-страницу, соответствующую приведенной ниже картинке. Использовать блочную вёрстку и CSS.

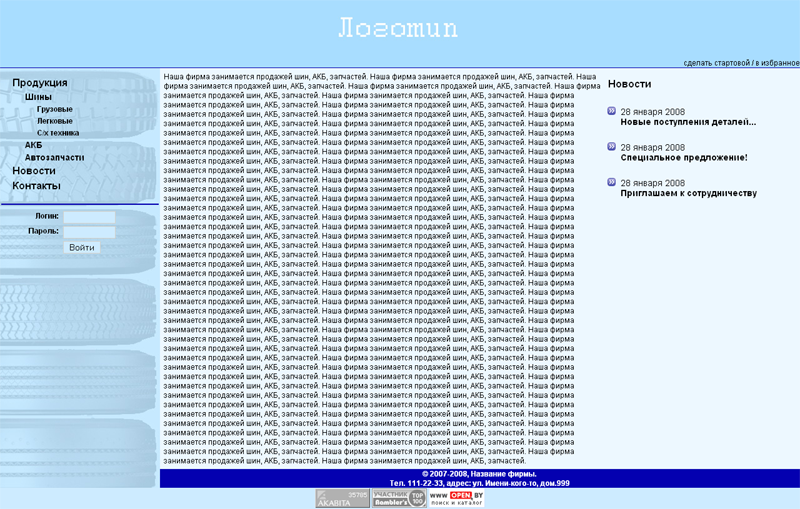


Рисунок 1.1 – Образец HTML-страницы

**Теоретические сведения**

Вёрстка веб-страниц — создание структуры HTML-кода, состоящей из программного кода на языках HTML, JavaScript, размещающей элементы веб-страницы (изображения, текст, стили оформления и т. д.) в окне браузера, согласно разработанному макету, таким образом, чтобы элементы дизайна выглядели аналогично макету.

Вёрстка веб-страниц отличается от полиграфической тем, что необходимо учитывать разницу отображения элементов в различных браузерах и разницу в размерах рабочего пространства устройств.

Процесс сложен и имеет творческую основу, ни один из способов не является каноничным и принятым как основа. Все подходы к вёрстке имеют как преимущества, так и недостатки.

Блочная вёрстка — верстка с помощью тега <div> и описывающего их множество таблицы стилей (CSS).

**Код программы:**

Index.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Лабораторная работа - 1</title>

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<div class="container">

<header class="header">

<a href="#" class="in-favorite">

сделать стартовой / в избранное

</a>

</header>

<aside class="aside">

<nav class="menu">

<a href="#" class="main">Продукция</a>

<a href="#" class="sub">Шины</a>

<a href="#" class="type">Грузовые</a>

<a href="#" class="type">Легковые</a>

<a href="#" class="type">С/х техника</a>

<a href="#" class="sub">АКБ</a>

<a href="#" class="sub">Автозапчасти</a>

<a href="#" class="main">Новости</a>

<a href="#" class="main">Контакты</a>

</nav>

<form action="" class="form">

<div class="row">

<div class="half">Логин:</div>

<div class="half">

<input type="text">

</div>

</div>

<div class="row">

<div class="half">Пароль:</div>

<div class="half">

<input type="text">

</div>

</div>

<div class="row">

<button>Войти</button>

</div>

</form>

</aside>

<main class="main-wrap">

<div class="content">

<iframe src="" frameborder="0"></iframe>

<div class="down-line">

<p>

@ 2007-2008, Название фирмы.

</p>

<p>

Тел. 111-22-33, адрес:ул. Имени кого-то,дом 999

</p>

</div>

</div>

<div class="news">

<h2 class="title">Новости</h2>

<div class="item">

<div class="ico">

</div>

<div class="date">

28 января 2008

</div>

<a href="#" class="theme">

Новые поступления деталей...

</a>

</div>

<div class="item">

<div class="ico">

</div>

<div class="date">

28 января 2008

</div>

<a href="#" class="theme">

Специальное предложение

</a>

</div>

<div class="item">

<div class="ico">

</div>

<div class="date">

28 января 2008

</div>

<a href="#" class="theme">

Приглашаем к сотрудничеству

</a>

</div>

</div>

</main>

<footer class="footer">

<a href="#">

<img src="img/image1.bmp" alt="">

</a>

<a href="#">

<img src="img/image1.gif" alt="">

</a>

<a href="#">

<img src="img/image3.bmp" alt="">

</a>

</footer>

</div>

</body>

</html>

text.html

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Наша фирма занимается продажей шин, АКБ, запчастей. Наша фирма занимается продажей шин, АКБ, запчастей.

Результат выполненной работы представлен на рисунке 1.2.

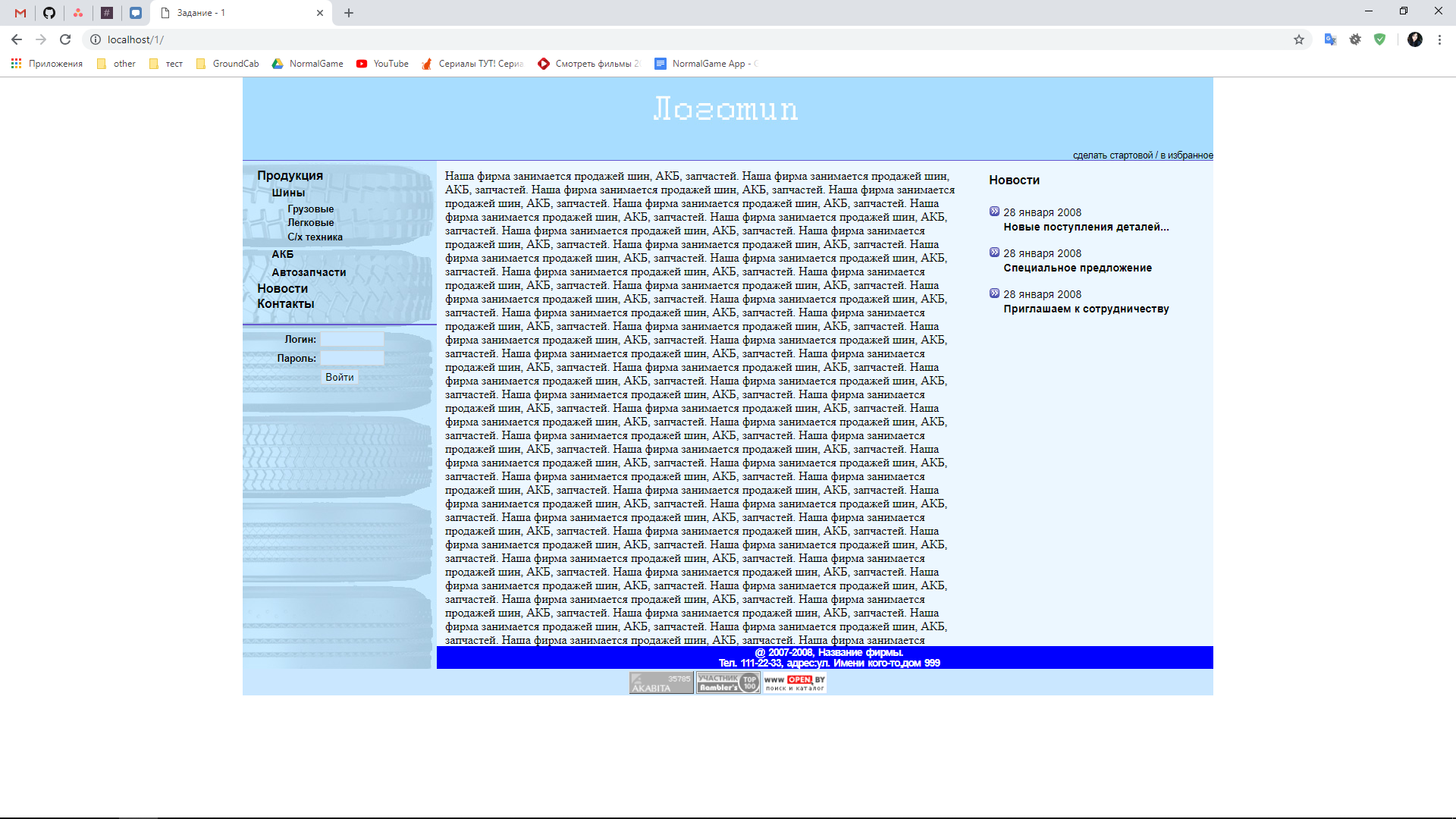


Рисунок 1.2 – Результат разработанной HTML-страницы

**Вывод**

Верстка веб-сайта осуществляется с применением двух основных способов: табличной и блочной верстки. Блочная верстка более функциональна и представляет собой больше возможностей благодаря использованию CSS.

Особенности:

– HTML-размеченный сайт является статическим сайтом.

Применение:

– HTML применяется при создании различных сайтов, интернет-магазинов, CMS-систем, helpdesks, CRM-систем, сайтов визиток и т.д.

**Задание 2. Основы PHP**

**Цель**: Изучить основы PHP, способы объявления переменных и вывода их на экран

**Задание:**

1. Объявить переменные следующих типов: целочисленную, строковую, дробную, логическую, массив.

2. Объявить переменные $a=555 и $b="ZZZ" и сложить их: а) как числа, б) как строки. Результат сложения не помещать в новую переменную, а сразу выводить на экран.

3. Есть три сотрудника:

a. Иванов, тел. 111-22-33, e-mail: ivanov@domain.com

b. Петров, тел. 112-24-36, e-mail: petrov@domain.com

c. Сидоров, тел. 113-25-37, e-mail: sidorov@domain.com

Объявить двухмерный массив, первый уровень которого пронумерован, начиная с нуля, а второй уровень содержит элементы name, phone, email, в которых хранятся соответствующие данные вышеназванных сотрудников.

4. Дан массив, содержащий элементы: 1, 2, "A", 3.764, 34, "B", 12. Объявить этот массив, проанализировать его содержимое и удалить из него все элементы, не являющиеся целыми или дробными числами.

5. Сгенерировать HTML-таблицу, состоящую из трёх колонок и 1000 строк. В первой колонке разместить номера строк таблицы. Цвет каждой строки таблицы должен изменяться по алгоритму: R+1, G+1, B+1, начиная с 000000. Т.е.: первая строка: 000000, вторая – 010101, третья -020202 и т.д. Помните, что числа – шестнадцатеричные и максимальное значение цвета: FFFFFF.

Задание требуется выполнить на операционной системе Linux.

**Теоретические сведения**

PHP является языком с динамической типизацией. Это значит, что тип данных переменной выводится во время выполнения, и в отличии от ряда других языков программирования в PHP не надо указывать перед переменной тип данных.

PHP поддерживает восемь простых типа данных:

– boolean (логический тип)

– integer (целые числа)

– double (дробные числа)

– string (строки)

– array (массивы)

– object (объекты)

– resource (ресурсы)

– NULL

Виды операторов в PHP: Унарные (работают только с одним аргументом, например ! – оператор отрицания), бинарные, и тернарный оператор ? (используется для условного выбора между двумя операторами)

PHP не требует (и не поддерживает) явного определения типа при объявлении переменной; тип переменной определяется по контексту, в котором она используется. То есть, если вы присваиваете строковое значение переменной var, var становится строкой. Если вы затем присваиваете переменной var целочисленное значение, она становится целым числом.

Результат выполненной работы представлен на рисунке 2.1, 2.2, 2.3

**Код программы:**

<?php

echo "<h1>Задание 1.</h1><br>";

echo php\_uname();

echo "<br><h1>Задание 2.</h1><br>";

$integer = 1;

$string = "text";

$double = 2.3;

$boolean = false;

$array = [2];

var\_dump($integer);

echo "<br>";

var\_dump($string);

echo "<br>";

var\_dump($double);

echo "<br>";

var\_dump($boolean);

echo "<br>";

var\_dump($array);

echo "<br><h1>Задание 3.</h1><br>";

$a = 555;

$b = 'ZZZ';

echo $a + $b;

echo "<br>";

echo $a . $b;

echo "<br><h1>Задание 4.</h1><br>";

$employers = [

[

"name" => "Иванов",

"phone" => "111-22-33",

"email" => "ivanov@domain.com"

],

[

"name" => "Петров",

"phone" => "112-24-36",

"email" => "petrov@domain.com"

],

[

"name" => "Сидоров",

"phone" => "113-25-37",

"email" => "sidorov@domain.com"

]

];

echo "<br>";

foreach ($employers as $index => $employer) {

echo "[$index] ";

foreach ($employer as $key => $info) {

echo "[$key] = $info ";

}

echo "<br>";

}

echo "<br><h1>Задание 5.</h1><br>";

$nums = [1, 2, "A", 3.764, 34, "B", 12];

$nums = array\_filter($nums, function ($num) {

return is\_numeric($num);

});

var\_dump($nums);

echo "<br><h1>Задание 6.</h1><br>";

echo '<table width="300">';

for ($row = 0; $row < 1000; $row++) {

echo "<tr style='background-color: #" . dechex($row % 256) . dechex($row % 256) . dechex($row % 256) . "'>";

for ($column = 0; $column < 3; $column++) {

echo '<td style="width:100">';

echo ($column == 0 ? $row : '');

echo "</td>";

}

echo "</tr>";

}

echo '</table>';

echo "<br><h1>Дополнительно задание.</h1><br>";

$data = [

[

"field" => "field1",

"value" => "value1",

],

[

"field" => "field2",

"value" => "value2",

],

[

"field" => "field3",

"value" => "value3",

],

[

"field" => "field4",

"value" => "value4",

],

[

"field" => "field5",

"value" => "value5",

],

[

"field" => "field6",

"value" => "value6",

],

[

"field" => "field7",

"value" => "value7",

],

[

"field" => "field8",

"value" => "value8",

],

];

echo '<form method="POST">';

foreach ($data as $item) {

echo '<input type="text" name="' . $item[field] . '" value="' . $item[value] . '"><br>';

}

echo '<input type="submit">';

echo "</form>";

if(!empty($\_POST))

{

foreach ($\_POST as $key => $value) {

$val = "\"$key\" = \"$value\"";

file\_put\_contents('data.txt', $val.PHP\_EOL , FILE\_APPEND | LOCK\_EX);

}

}

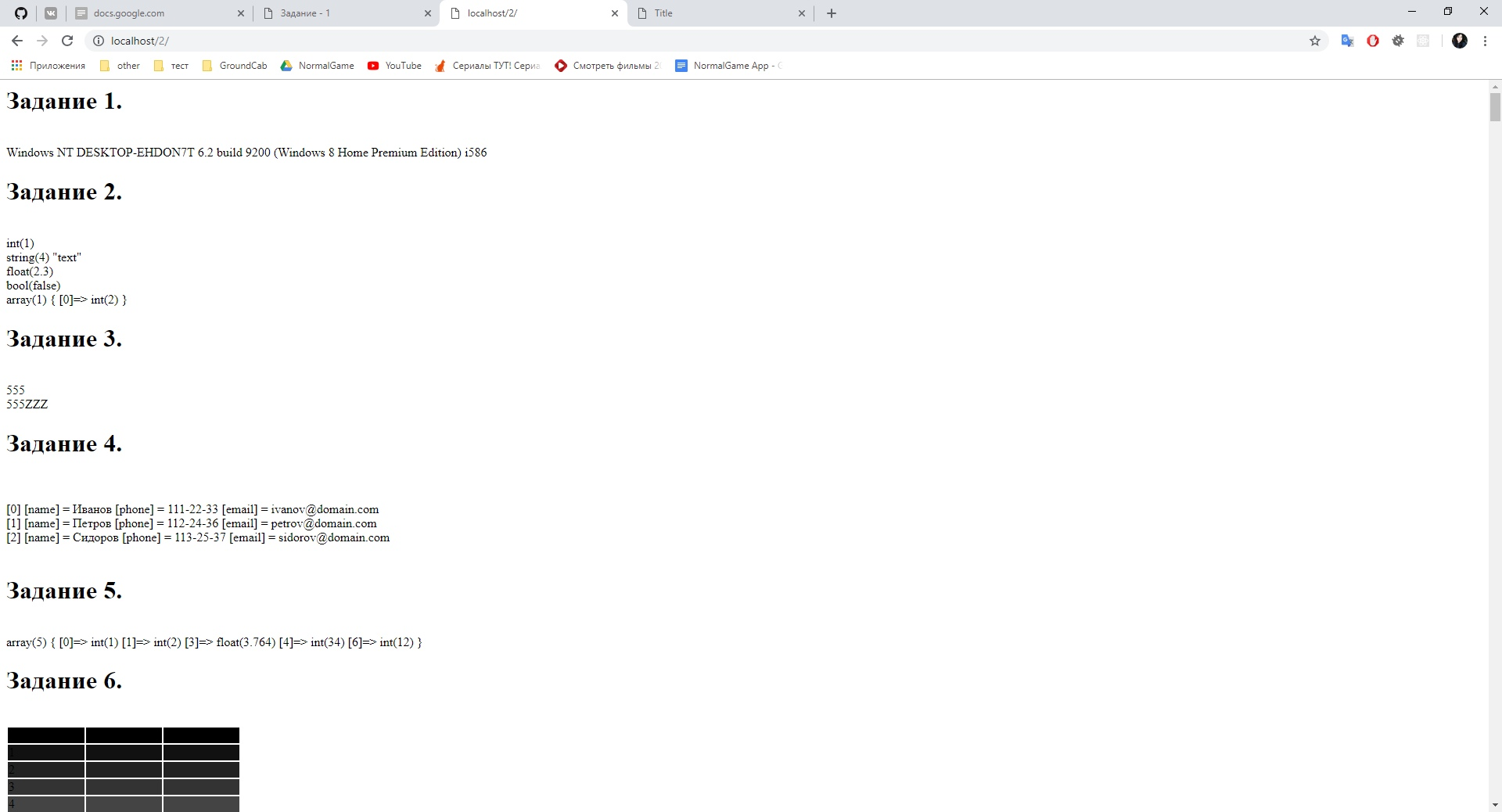


Рисунок 2.1 – Результат вывода задания 1-6

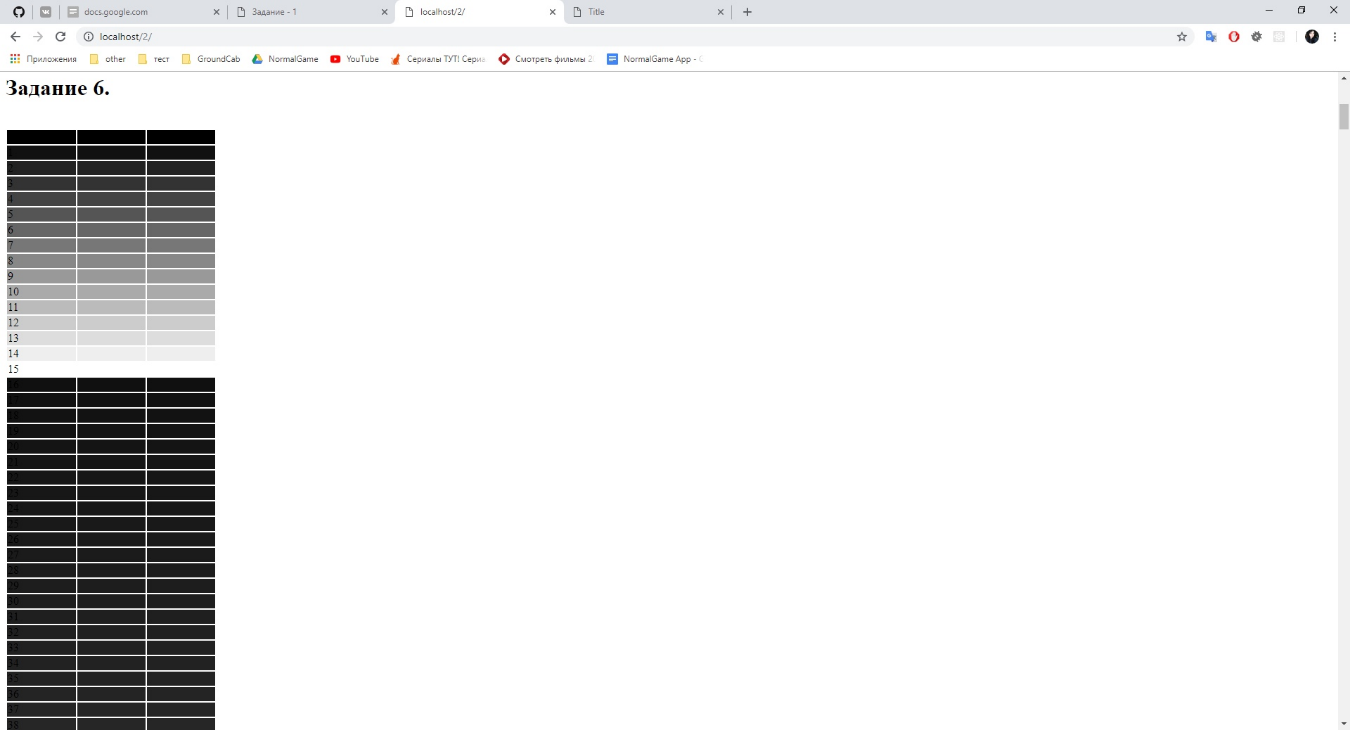


Рисунок 2.2 – Результат вывода 6 задания

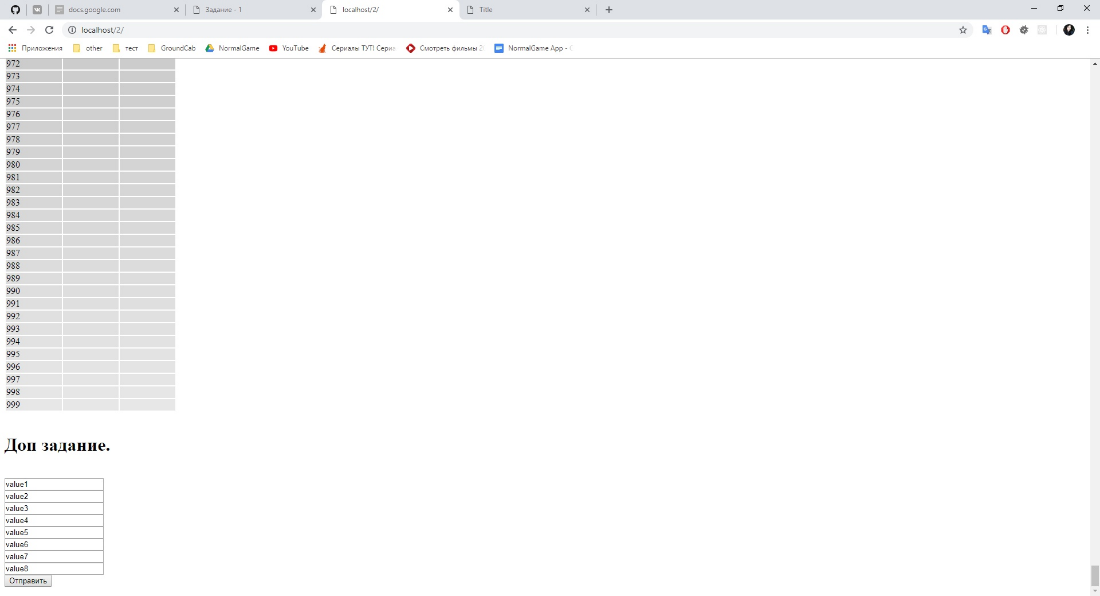


Рисунок 2.3 – Результат вывода дополнительного задания

**Вывод**

В результате работы были изучены теоретические сведения о языке PHP и были выполнены все задания в соответствии с требованиями.

Применение:

Главная область применения PHP – написание скриптов, работающих на стороне сервера; таким образом, PHP способен выполнять все то, что выполняет любая другая программа CGI, например обрабатывать данные форм, генерировать динамические страницы, принимать и отсылать cookies и др.

Особенности:

PHP способен генерировать не только HTML. Доступно формирование изображений, файлов PDF и даже роликов Flash (с использованием libswf и Ming), создаваемых «на лету».

**Задание 3. Основы разработки сайтов**

**Цель:** Изучить способы создания сайтов с помощью HTML шаблонов. Изучить возможности использования плейсхолдеров. В частности изучить использование шаблонов с использованием шаблонов .tpl. Изучить разделение кода и дизайна с использование tpl шаблонов.

**Задание:**

1. Прочитать содержимое шаблона main.tpl в текстовую переменную.

2. Плейсхолдер {MAIN\_MENU} заменить на содержимое шаблона main\_menu.tpl

3. Блоки: а) с основным текстом страницы, б) новостями, в) областью с адресом и копирайтом удалить из шаблона main.tpl, пометив места их расположения плейсхолдерами. Содержимое удаляемых блоков разместить в отдельных шаблонах. Обеспечить замену плейсхолдеров содержимым соответствующих шаблонов (аналогично пункту 2).

4. Плейсхолдеры {TODAY\_D}, {TODAY\_M}, {TODAY\_Y}, {NOW\_H}, {NOW\_M}, {NOW\_S} заменить на фрагменты текущей даты и времени: день, месяц, год, час, минута, секунда соответственно.

5. Создать конфигурационный файл site.cfg (формат разработать самостоятельно; примечание: делать этот файл в формате PHP ЗАПРЕЩЕНО, т.е. это должен быть "простой текстовый файл"), в котором разместить две переменные: main\_color и copyright\_color, в которых хранить (и использовать для управления сайтом) цвета области с основным текстом страницы и области с адресом и копирайтом.

6. Дополнительное задание (по желанию): разработать формат файла для хранения новостей, перенести туда все новости, написать функцию, формирующую список новостей на основе этого файла и шаблонов, определяющих отображение одной отдельной новости.

7. Дополнительное задание (по желанию): меню сайта расположить в отдельном файле (формат разработать самостоятельно) в виде иерархической структуры. Прочитать этот файл в многомерный массив. На основе полученного массива сформировать главное меню сайта.

**Теоретические сведения**

Файлы tpl используются для написания в них HTML-кода, который впоследствии будет подключаться в PHP и выведен пользователю на конкретных страницах. Например, в файле tpl может быть форма регистрации. То есть просто HTML-код обычной формы, ничего больше. В другом файле tpl может быть блок с панелью пользователя.

В файлах tpl используются специальные элементы - шаблонизаторы. Они записываются примерно в таком виде: {username}.

Страница будет собираться по кирпичикам, выводя все tpl-файлы в нужном порядке. Главное преимущество такого подхода, что разделяется программная часть и дизайнерская, благодаря чему становится крайне легко изменять дизайн и код PHP независимо друг от друга.

**Код программы:**

index.php:

<?php

$main = file\_get\_contents("templates/main.tpl");

$config = json\_decode(file\_get\_contents("site.cfg"), true);

date\_default\_timezone\_set("Europe/Minsk");

$replace = [

"main\_menu" => file\_get\_contents("templates/main\_menu.tpl"),

"today\_d" => date("d"),

"today\_m" => date("m"),

"today\_y" => date("y"),

"now\_h" => date("H"),

"now\_m" => date("i"),

"now\_s" => date("s"),

"news" => file\_get\_contents("templates/news.tpl"),

"main\_text" => file\_get\_contents("templates/main\_text.tpl")

];

$replace += $config;

$main = str\_replace(array\_map(function($item){return "{".strtoupper($item)."}";}, array\_keys($replace)), $replace, $main);

echo $main;

site.cfg

MAIN\_COLOUR=yellow

COPYRIGHT\_COLOR=green

HEADER\_TITLE=Задание-3

main.tpl

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html lang="ru" xml:lang="ru" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="content-type" content="text/html;charset=UTF-8" />

<title>{HEADER\_TITLE}</title>

<META content="{HEADER\_DESCRIPTION}" name="description">

<META content="{HEADER\_KEYWORDS}" name="keywords">

<link href="css/basic.css" rel="stylesheet" type="text/css" media="all" />

</head>

<body>

<table width="100%" border="0" bgcolor="#a8ddff" class="globaltable" cellspacing="0" cellpadding="0">

<tr class="globaltable\_ur">

<td colspan="3" align="center" valign="top">

<img src="images/logo.jpg">

</td>

</tr>

<tr>

<td colspan="3"align="right">

<a href=# onClick="this.style.behavior='url(#default#homepage)'; this.setHomePage('http://www.somedomain.by/'); return false;">сделать стартовой</a>&nbsp;/&nbsp;<a rel="sidebar" onclick="window.external.AddFavorite(location.href,'ФИРМА'); return false;" href="http://www.somedomain.by/" title="ФИРМА">в избранное</a>

</td>

</tr>

<tr bgcolor="#0400b0" height="1">

<td colspan="3"></td>

</tr>

<tr>

<td align="left" valign="top" width="20%" class="menutd" rowspan="2">

Сейчас: {TODAY\_D}.{TODAY\_M}.{TODAY\_Y} {NOW\_H}:{NOW\_M}:{NOW\_S}

<br /><br />

{MAIN\_MENU}

<form action="#" method="post">

<table border="0" width="100%" align="center">

<tr bgcolor="#0400b0" height="1">

<td colspan="2"></td>

</tr>

<tr height="5">

<td colspan="2"></td>

</tr>

<tr>

<td align="right" valign="middle"><b>Логин:</b>&nbsp;</td>

<td align="left"><input type="text" size="10" name="ulogin" class="formf1" /></td>

</tr>

<tr>

<td align="right" valign="middle"><b>Пароль:</b>&nbsp;</td>

<td align="left"><input type="password" size="10" name="upassword" class="formf1" /></td>

</tr>

<tr>

<td>&nbsp;</td>

<td align="left"><input type="submit" name="go" value="Войти" class="formf1" /></td>

</tr>

</table>

</form>

</td>

<td width="80%" bgcolor="#edf8ff">

<!-- inner table for main only -->

<table border="0" width="100%" cellspacing="5" cellpaddong="5">

<tr>

<td width="70%" style="background-color:{MAIN\_COLOUR}">

{CONTENT}

</td>

<td width="30%">

{NEWS}

</td>

</tr>

</table>

</td>

</tr>

<tr>

<td align="center" width="80%" style="background-color:{COPYRIGHT\_COLOR}" class="bottomline">

{FOOTER}

</td>

</tr>

<tr>

<td align="center" valign="top" bgcolor="#cdebff" width="100%" colspan="2">

<img src="images/1.bmp"> <img src="images/2.gif"> <img src="images/3.bmp">

</td>

</tr>

</table>

</body>

</html>

Результат выполненной работы представлен на рисунке 3.1

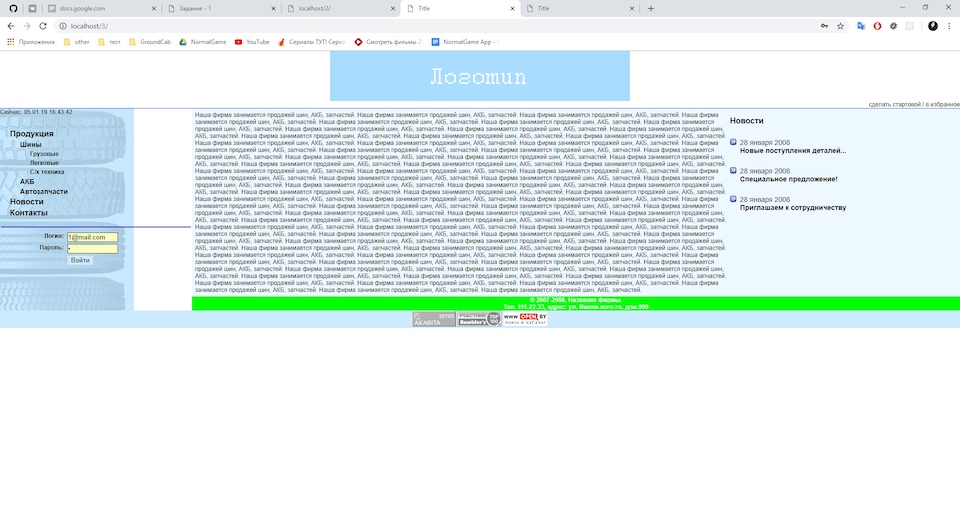


Рисунок 3.1 – Результат разработки страницы

**Вывод**

В ходе выполнения работы были изучены и применены на практике методы работы с шаблонами PHP.

Использование шаблонов при создании сайта помогает сэкономить время разработчиков, т.к. обеспечивает разделение логики работы приложения от способа представления данных т.е. дизайна, что делает код понятным и удобным для восприятия. Удобно использовать при работе в команде.

Особенности:

Когда часть информации полностью дублируется на нескольких страницах ее можно положить в один файл и использовать при генерации всех этих страниц.

Применение:

Генерации страниц с одной и той же информацией, но различным оформлением – например, обычная версия и версия для печати.

**Задание 4. Основы взаимодействия с БД**

**Цель:** Изучить основы взаимодействия PHP с базами данных.

**Задание:**

1. Создать БД с уникальным именем.

2. В созданной БД создать таблицу news, содержащую поля: уникальный идентификатор новости, дата публикации новости, заголовок новости, текст новости.

3. Написать функцию, отображающую на сайте N самых свежих новостей (N задаётся как параметр функции), новости сортируются по убыванию даты. Рекомендуется для выполнения этого задания использовать шаблоны.

4. В созданной БД создать таблицу pages, содержащую поля: уникальный идентификатор страницы, информация о родительской странице, текст для отображения в меню.

5. На основе таблицы, разработанной в пункте 4, сформировать меню сайта. Меню должно иметь не менее трёх уровней вложенности. (Необязательное требование: при клике по некоторому пункту меню должен открываться ("разворачиваться") его первый подуровень, остальные пункты меню при этом находятся в "свёрнутом" состоянии).

6. Дополнительное задание (по желанию): добавить в таблицу pages поле «текст страницы» и модифицировать свой скрипт так, чтобы по клику на пункт меню в основной области отображался текст соответствующей страницы.

7. Дополнительное задание (по желанию): создать таблицу для хранения информации о пользователях сайта. Получить информацию из полей «Логин» и «Пароль» (форма на странице уже есть) и, если пользователь ввёл верные данные, заменить форму на текст в виде «Здравствуйте, ИМЯ\_ПОЛЬЗОВАТЕЛЯ!». Используя механизм сессий, добиться того, чтобы пользователь оставался «залогиненным» при переходе между различными страницами вашего сайта.

**Теоретические сведения**

База данных (БД) — это организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации, преимущественно больших объемов. Базы данных активно используются для динамических сайтов со значительными объемами данных — часто это интернет-магазины, порталы, корпоративные сайты. Такие сайты обычно разработаны с помощью серверного языка программирования (как пример, PHP) или на основе CMS (как пример, WordPress), и не имеют готовых страничек с данными по аналогии с HTML-сайтами. Странички динамических сайтов формируются «на лету» в результате взаимодействия скриптов и баз данных после соответствующего запроса клиента к веб-серверу.

В контексте баз данных стоит рассмотреть понятие СУБД. Система управления базами данных (СУБД) — это комплекс программных средств, необходимых для создания структуры новой базы, ее наполнения, редактирования содержимого и отображения информации. Наиболее распространенными СУБД являются MySQL, PostgreSQL, Oracle, Microsoft SQL Server.

Расширение для PHP, предоставляющее разработчику простой и универсальный интерфейс для доступа к различным базам данных - PHP Data Objects (PDO).

Результат работы программы представлен на рисунке 4.1

**Код программы:**

Index.php:

<?php

header('Content-type: text/html; charset=utf-8');

date\_default\_timezone\_set("Europe/Minsk");

include\_once "ViewHelper.php";

$viewHelper = new ViewHelper();

echo $viewHelper->getTemplate();

DBHelper.php:

<?php

class DBHelper

{

private $mysql;

public function \_\_construct()

{

$db = json\_decode(file\_get\_contents("db.cfg"), true);

$this->mysql = new mysqli($db['host'], $db['user'], $db['password'], $db['database']);

$this->mysql->set\_charset("utf8");

if ($this->mysql->connect\_error) {

die('Connection error');

}

}

public function getNews($count = 3)

{

$count = (int)$count ? (int)$count : 3;

$result = $this->mysql->query("SELECT \* FROM news ORDER BY date DESC LIMIT $count");

$news = [];

while ($data = $result->fetch\_assoc()) {

$news[] = $data;

}

return $news;

}

public function getMenu()

{

$menu = [];

$result = $this->mysql->query("SELECT \* FROM pages");

while ($data = $result->fetch\_assoc()) {

$menu[] = $data;

}

return $this->formatMenu($menu);

}

private function formatMenu($menu, $parentId = null)

{

$result = [];

foreach ($menu as $menuItem) {

if ($parentId == $menuItem["parent\_id"]) {

$result[] = $menuItem;

if ($submenu = $this->formatMenu($menu, $menuItem['id'])) {

$result[count($result) - 1]['submenu'] = $submenu;

}

}

}

return $result;

}

}

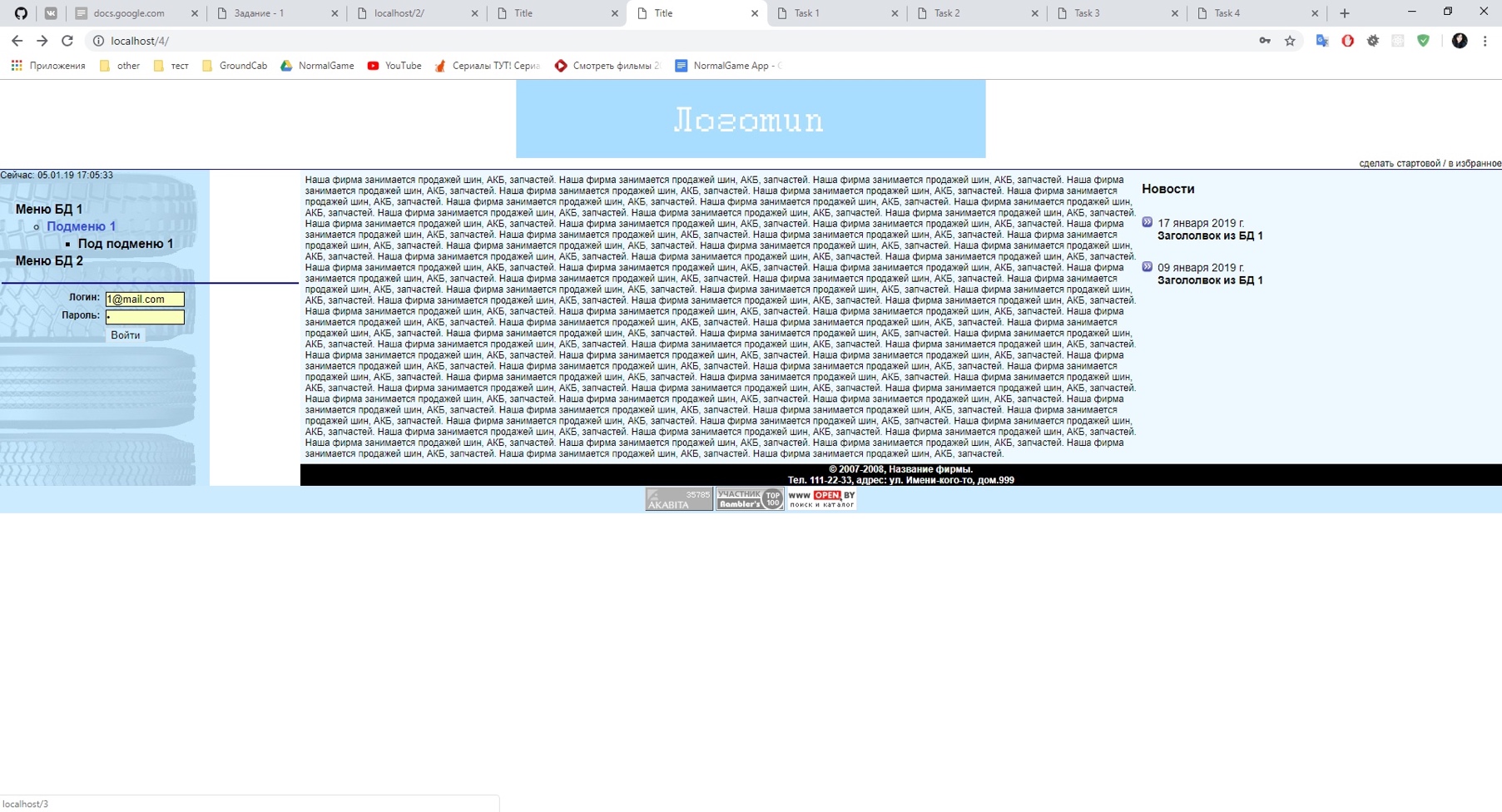


Рисунок 4.1 – Результат разработки страницы

**Вывод**

В ходе работы была создана база данных, в которой хранятся новости и пункты меню сайта.

Особенности:

* данные хранятся в таблицах, состоящих из столбцов и строк;
* на пересечении каждого столбца и строчки стоит в точности одно значение;
* у каждого столбца есть свое имя, которое служит его названием, и все значения в одном столбце имеют один тип; например, в столбце id все значения имеют целочисленный тип, а в строке name – текстовый;
* столбца располагаются в определенном порядке;

Запросы к базе данных возвращают результат в виде таблиц, которые тоже могут выступать как объект запросов.

Применение:

* в организациях для учета персонала, ведения бухгалтерии, учета товаров на складе, поставщиков, партнеров, клиентов. ведения электронного документооборота и т.д.
* в адресных и телефонных книгах, словарях справочниках;
* в биллинговых системах для учета траффика у Интернет-провайдеров, потребленных услуг у телефонных операторов, в банковском деле;
* в веб-приложениях;

**Задание 5. Основы работы с регулярными выражениями.**

**Цель:** Изучить правила работы с регулярными выражениями и применить их.

**Задание:**

1. Написать регулярное выражение, проверяющее, является ли строка корректным e-mail адресом.

Примеры корректных адресов:

* vasya-pupkin@mail.com
* vasya\_pupkin@mail.com
* vasya.pupkin@mail.com
* v.v.pupkin@firma.mail.com
* v.v.pupkin@firma-mail.com
* v.v.pupkin12@firma\_mail.com
* v.v.pupkin-director@firma.mail.com

Примеры некорректных адресов:

* -vasya--pupkin@mail.com
* vasya\_pupkin@mail..com
* vasya.-pupkin@mail.com
* v.v.pup kin@firma.mail.com
* v.v.pup#kin@firma-mail.com\_

2. Написать регулярное выражение, удаляющее из текста HTML-комментарии, т.е. конструкции вида

<!-- комментарий -->

При этом учесть, что конструкции вида

<!--[if IE]>

<link href="/css/invstroyIEfix.css" rel="stylesheet" type="text/css" />

<![endif]-->

удалять не надо, т.к. это, фактически, не комментарий.

3. Написать регулярное выражение, очищающее текст от HTML-тегов.

4. Написать регулярное выражение, выделяющее красным жирным шрифтом все слова с длиной в 5 и более символов, состоящие только из заглавных букв.

5. Дополнительное задание (по желанию): написать скрипт, представляющий HTML-документ в виде дерева элементов (дерево хранить в массиве).

**Теоретические сведения**

Регулярные выражения — формальный язык поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании метасимволов. Для поиска используется строка-образец (pattern), cостоящая из символов и метасимволов и задающая правило поиска. Для манипуляций с текстом дополнительно задаётся строка замены, которая также может содержать в себе специальные символы.

Основой синтаксиса регулярных выражений является тот факт, что некоторые символы встречающиеся в строке рассматриваются не как обычные символы, а как имеющие специальное значение (т.н. метасимволы). Именно это решение позволяет работать всему механизму регулярных выражений. Каждый метасимвол имеет свою собственную роль.

Вот примеры основных метасимволов:

**^**     - (крышка, цирркумфлекс) начало проверяемой строки

**$**     - (доллар) конец проверяемой строки

**.**     - (точка) представляет собой сокращенную форму записи для символьного класса, совпадающего с любым символом

**|**     -  означает «или». Подвыражения, объединенные этим способом, называются альтернативами (alternatives)

**?**     - (знак вопроса) означает, что предшествующий ему символ является необязательным

**+**     -  обозначает «один или несколько экземпляров непосредственно предшествующего элемента

**\***     –  любое количество экземпляров элемента (в том числе и нулевое)

**\\d**   –  цифровой символ

**\\D**   –  не цифровой символ

**\\s**   –  пробельный символ

**\\S**   –  не пробельный символ

**\\w**   –  буквенный или цифровой символ или знак подчёркивания

**\\W**   –  любой символ, кроме буквенного или цифрового символа или знака подчёркивания

**Вывод**

Регулярные выражения состоят из потерна и необязательных флагов. Основа регулярного выражения – патерн. Это строка, которую можно расширить специальными символами, которые делают поиск намного проще.

Особенности:

Регулярные выражения – формальный язык поиска. С помощью его можно найти подстроку в тексте с помощью специальной маски, построенной с помощью мета-символов.

Применение:

Применяется при разработке множества приложений, когда необходим быстрый и сложный поиск подстроки в тексте.

**Код программы:**

task1.html:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>Task 1</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<script src="task1.js"></script>

</head>

<body>

<input type="text" value="" id="email">

<br>

<button id="check">Check valid</button>

<br>

<hr>

<span id="result"></span>

<br>

</body>

</html>

task2.html:

<html>

<head>

<meta charset="utf-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>Task 2</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<script src="task2.js"></script>

</head>

<body>

<button id="remove-comments">Remove comments</button>

<xmp id="before">

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<!--[if IE]>

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<![endif]-->

<title>Page Title</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" type="text/css" media="screen" href="main.css" />

<script src="main.js"></script>

</head

<body>

<!--comment1-->

<!-- comments2 -->

</body>

</html>

</xmp>

<xmp id="after"></xmp>

</body>

</html>

**Задание 6. Основы работы с лентами новостей.**

**Цель:** Изучить и предоставить новости в формате RSS.

**Задание:** Представить новости в RSS формате.

**Теоретические сведения**

Новостная лента — формат данных, используемый для доставки пользователям часто обновляемой информации. Распространители этой информации предоставляют новостную ленту, позволяя пользователям подписаться на неё.

Лента состоит из некоторого ограниченного числа статей, а также из служебной информации типа автора ленты или домашней странички. При появлении новых данных они добавляются в ленту, вытесняя при этом старые статьи. Обычно в ленте бывает не больше 10-20 статей, хотя бывают ленты и с несколькими сотнями статей.

Новостные ленты автоматически собираются новостными агрегаторами, которые могут быть либо веб-приложениями, либо устанавливаться на компьютер конечного пользователя.

При помощи лент новостей может доставляться обычный HTML-контент (веб-страницы), ссылки либо другие типы цифровых данных. Например, в статьях в ленте, посвящённой какому-нибудь сериалу или подкасту может присутствовать особым образом сформированная ссылка, при обнаружении которой поддерживающие эту технологию агрегаторы скачают соответствующий файл. Таким образом реализуется периодическая доставка цифровых медиафайлов, также известная как Broadcatching.

Результат работы представлен на рисунке 6.1

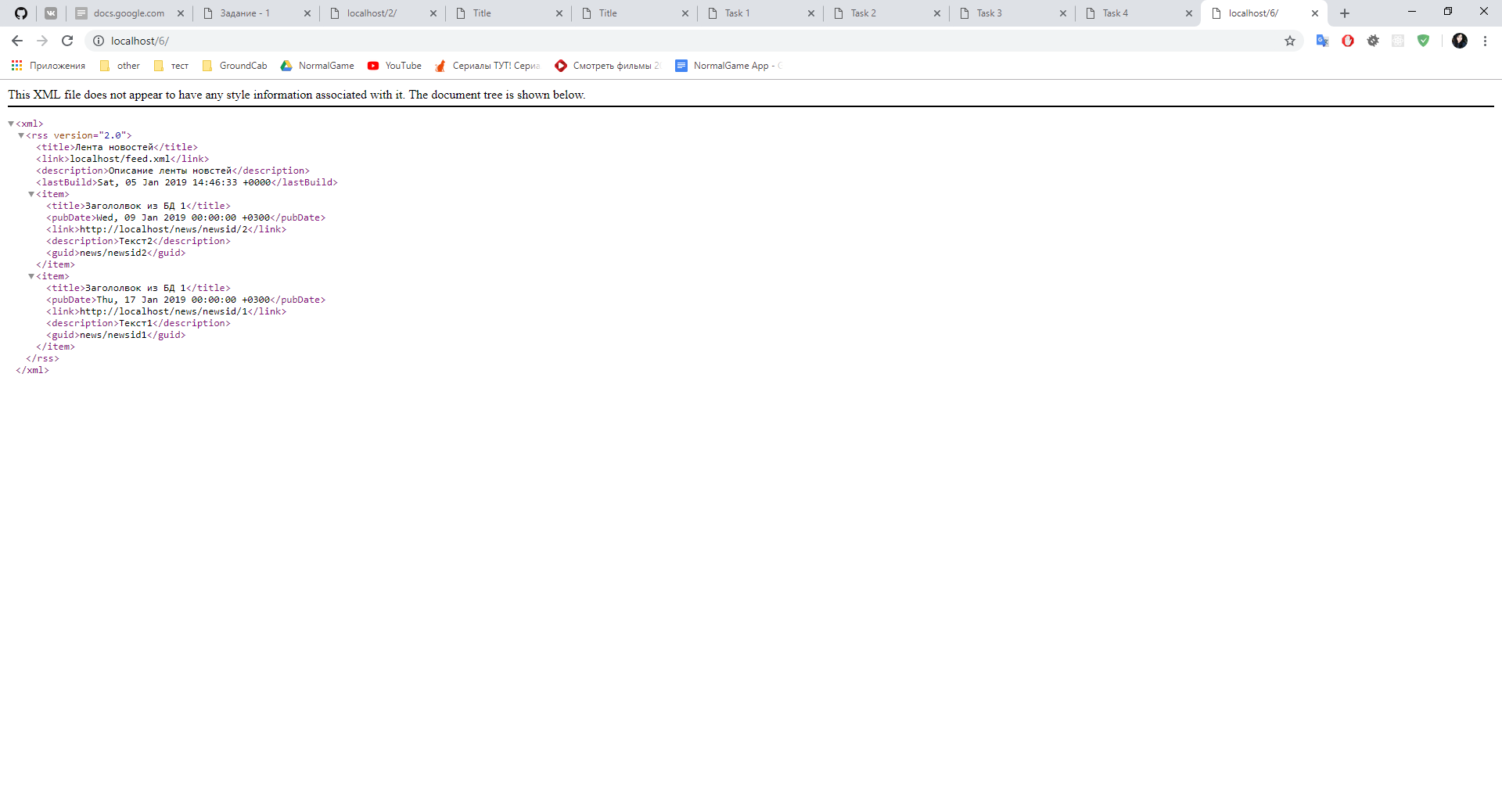


Рисунок 6.1 – Результат

**Вывод**

Были изучены основы работы с RSS лентами новостей

Транслируемые данные хранятся в БД, который расположен на хостинге владельца. Программы-клиенты загружают эти данные в автоматическом режиме с помощью PHP скрипта: как правило, для загрузки доступны заголовки и краткие аннотации новостей.

Применение:

Информация из различных источников, представленная в формате RSS, может быть собрана, обработана и представлена пользователю в удобном для него виде специальными программами агрегаторами или онлайн-сервисами. RSS удобно использовать, когда речь идет о сборе информации из нескольких источников одновременно: достаточно следить за появлением новостей на всех интересующих сайтах, не посещая их. Таким образом, можно сэкономить время, поскольку сбором и доставкой нужных сведений будет заниматься специальная программа.

**Код программы:**

**index.php:**

<?php

header("Content-Type: text/xml");

header("Expires: Thu, 19 Feb 1998 13:24:18 GMT");

header("Last-Modified: " . gmdate("D, d M Y H:i:s") . " GMT");

header("Cache-Control: no-cache, must-revalidate");

header("Cache-Control: post-check=0,pre-check=0");

$db = json\_decode(file\_get\_contents("db.cfg"), true);

$mysql = new mysqli($db['host'], $db['user'], $db['password'], $db['database']);

$mysql->set\_charset("utf8");

$news = [];

$result = $mysql->query("SELECT

title,

date,

CONCAT('http://" . $\_SERVER["HTTP\_HOST"] . "/news/newsid/', id) AS link,

body AS description,

CONCAT('news/newsid', id) AS guid

FROM news ORDER BY date ASC");

while ($data = $result->fetch\_assoc()) {

$news[] = $data;

}

$xml = new SimpleXMLElement("<xml/>");

$rss = $xml->addChild("rss");

$rss->addAttribute("version", "2.0");

$rss->addChild("title", "Лента новостей");

$rss->addChild("link", $\_SERVER["HTTP\_HOST"] . "/feed.xml");

$rss->addChild("description", "Описание ленты новстей");

$rss->addChild("lastBuild", gmdate("r"));

foreach ($news as $newsItem) {

$item = $rss->addChild("item");

foreach ($newsItem as $key => $newsPartItem) {

if ($key == "date") {

$date = (new DateTime($newsPartItem))->format("r");

$item->addChild("pubDate", $date);

} else {

$item->addChild($key, $newsPartItem);

}

}

}

echo $xml->asXML();