

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ И ФИЗИКО-МАТЕМАТИЧЕСКИХ НАУК
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ

ОТЧЕТ
по учебной практике №2

студента 2 курса
группа ПИБ-2301-01-00
Багаева Ильи Владимировича
Ф. И. О.

Руководитель учебной практики
Стариков Андрей Иванович
Ф. И. О.

Дата начала учебной практики: 19 июня 2017 г.
Дата окончания учебной практики: 7 июля 2017 г.

Оценка: _____

Киров
2017

Оглавление

Введение.....	3
Задание на учебную практику.....	4
Разработка веб-сайта. Практические результаты	5
I. Совершенствование умения в разметке веб-страницы с помощью тегов HTML	5
II. Приобретение навыков по оформлению веб-страницы с помощью таблицы стилей CSS	5
III. Рисование макета сайта	6
IV. Вёрстка сайта	7
V. Знакомство со скриптовым языком PHP	14
VI. Знакомство с CMS WordPress, изучение соответствующей документации	14
VII. Установка сайта на CMS WordPress	16
Заключение	25
Список использованных источников	25

Введение

Актуальность прохождения учебной практики заключается в непосредственном знакомстве с процессом разработки сайта; приобретением практических навыков по специальности.

Учебная практика способствует реализации теоретических знаний, умений и навыков; развитию навыков в разметке веб-страницы с помощью тегов HTML, прикреплении таблицы стилей CSS, установке сайта на CMS WordPress. А также получению представления о практической деятельности по специальности.

Целью прохождения учебной практики является совершенствование навыков создания веб-сайтов. Для достижения поставленных целей при прохождении практики ставились следующие задачи:

- 1) совершенствовать умения в разметке веб-страницы с помощью тегов HTML;
- 2) приобретение навыков по оформлению веб-страницы с помощью таблицы стилей CSS;
- 3) установка сайта на CMS WordPress.

Основная часть моей работы над заданием в период прохождения учебной практики составляла: рассмотрение и анализ материалов задания и дополнительных материалов, способствующих углублению знаний, необходимых для достижения поставленных целей, изучение тематической документации (учебные пособия, Интернет-ресурсы и документация по CMS WordPress), разработка веб-сайта и составление презентации, показывающей особенности его функционирования.

Задание на учебную практику

План практики включает в себя:

1. Рисование макета сайта
2. Вёрстка сайта
3. Установка сайта на CMS WordPress
4. Оформление отчёта и презентации

Разработка веб-сайта. Практические результаты

Для удобства процесс выполнения учебной практики был разбит на ряд задач:

1. Совершенствование умения в разметке веб-страницы с помощью тегов HTML
2. Приобретение навыков по оформлению веб-страницы с помощью таблицы стилей CSS
3. Рисование макета сайта
4. Вёрстка сайта
5. Знакомство со скриптовым языком PHP
6. Знакомство с CMS WordPress, изучение соответствующей документации
7. Установка сайта на CMS WordPress

I. Совершенствование умения в разметке веб-страницы с помощью тегов HTML

Для успешного выполнения задания учебной практики мною была изучена различная документация о языке разметки HTML. Были изучены основы языка HTML: что такое парные и одинарные теги, структура HTML страницы, кодировка и мета-теги, разметка и форматирование текста, создание ссылок, использование изображений, таблиц и форм.

Также было совершено знакомство с новыми возможностями HTML5 – новой открытой платформой, предназначенной для создания веб-приложений, использующих аудио, видео, графику, анимацию и многое другое.

HTML используется для структурирования содержимого страницы.

II. Приобретение навыков по оформлению веб-страницы с помощью таблицы стилей CSS

Для реализации презентабельного внешнего вида веб-страниц, мною был изучен язык CSS – формальный язык описания внешнего вида документа, написанного с использованием языка разметки. CSS – это язык стилей, определяющий отображение HTML-документов. Например, CSS работает с шрифтами, цветом, полями, строками, высотой, шириной, фоновыми изображениями, позиционированием элементов и многими другими вещами.

HTML может использоваться для оформления веб-сайтов. Но CSS предоставляет большие возможности, он более точен и проработан. CSS, на сегодняшний день, поддерживается всеми браузерами.

CSS используется для форматирования этого структурированного содержимого.

Изучив CSS, я выделил его некоторые преимущества:

- управление отображением множества документов с помощью одной таблицы стилей;

- более точный контроль над внешним видом страниц;
- различные представления для разных носителей информации (экран, печать, и т. д.);
- сложная и проработанная техника дизайна.

III. Рисование макета сайта

Разметка макета сайта была сделана в программе Adobe Photoshop. Для данных задач в данном ПО имеется огромное количество инструментов. Направляющие и сетка помогают точно расположить изображения или элементы. Направляющие представляют собой непечатаемые линии, покрывающие изображение.

Использование направляющих необходимо чтобы все элементы сайта были выровнены, а не "налезали" и "прятались" друг за другом. Направляющие в дальнейшем значительно облегчат верстку сайта.

После разметки начинается конструирование самого макета. В процессе работы были обработаны различные изображения, разработан макет шапки сайта, меню, ротатор изображений, блок основного контента главной страницы. Кроме того, был добавлен поиск по сайту, правый блок с категориями постов. Также созданы посты, которые можно будет удобно читать, сделан футер, который не будет отвлекать от контента, но будет давать всю необходимую информацию о сайте.

Фрагмент готового макета, созданного в программе Adobe Photoshop представлен на рисунке 1.

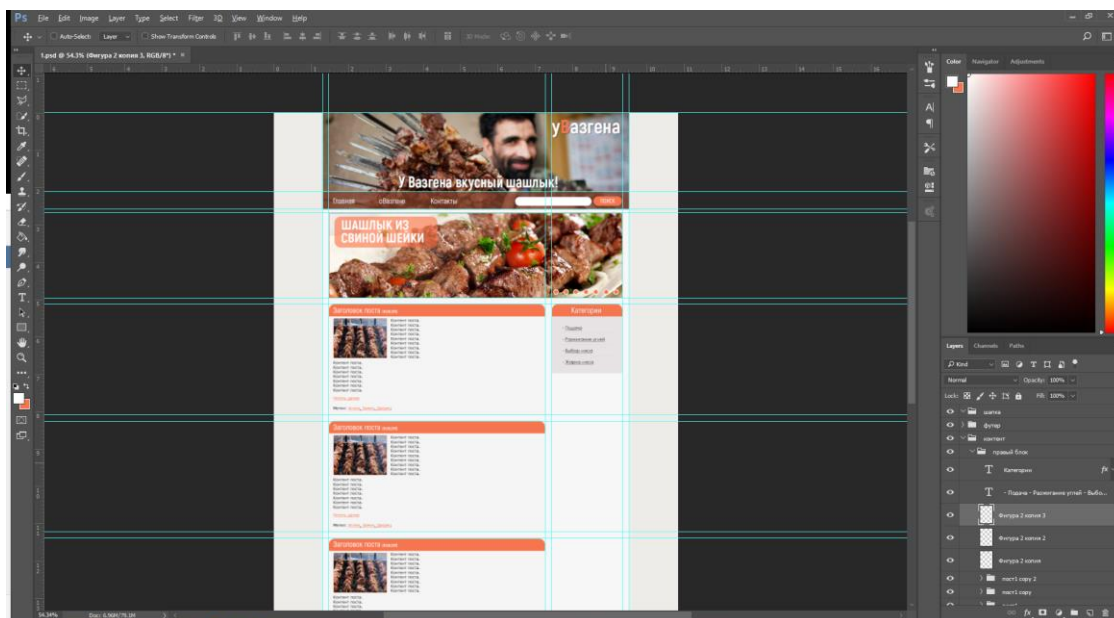


Рисунок 1 – Макет веб-страницы в программе Adobe Photoshop

IV. Вёрстка сайта

Для успешного отображения веб-страницы, необходимо хорошо продумать разметку будущей HTML-страницы. Необходимо выделить основные блоки, из которых она будет состоять:

- «шапка» сайта (включая меню и ротатор изображений);
- блок основного контента (включая раздел с категориями постов);
- футер («подвал») (включая все входящие в него списки, меню).

Первым делом идёт вёрстка «шапки» сайта, меню и формы для поиска. Вместе с этим устанавливается логотип и слоган сайта. После чего необходимо сверстать область основного контента, навигацию по категориям и «подвал» сайта. Шапка и первый пост сверстанной HTML-страницы (без использования таблицы стилей CSS) представлены на рисунке 2.



Рисунок 2 – «Шапка» и часть блока основного контента HTML-страницы

На рисунке 3 представлены «подвал» и оставшаяся часть блока основного контента HTML-страницы.

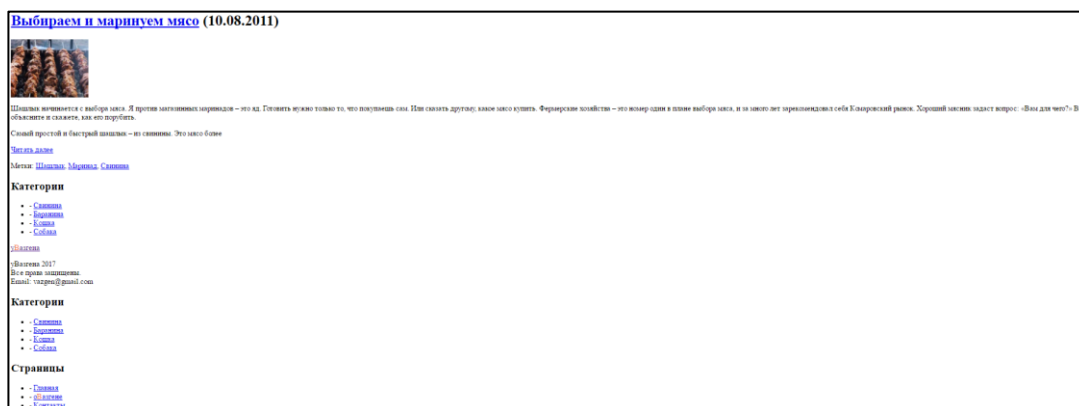


Рисунок 3 – «Подвал» и часть блока основного контента HTML-страницы

Закончив создание HTML-разметки, я приступил к оформлению сайта с помощью таблицы стилей CSS. Чтобы уменьшить количество используемых изображений и не увеличивать вес страницы, при верстке использовались новые свойства CSS. Также, чтобы уменьшить количество используемых классов, активно применялись псевдо-классы. На рисунке 4 представлены шапка и часть блока основного контента сверстанной HTML-страницы с использованием стилей CSS.

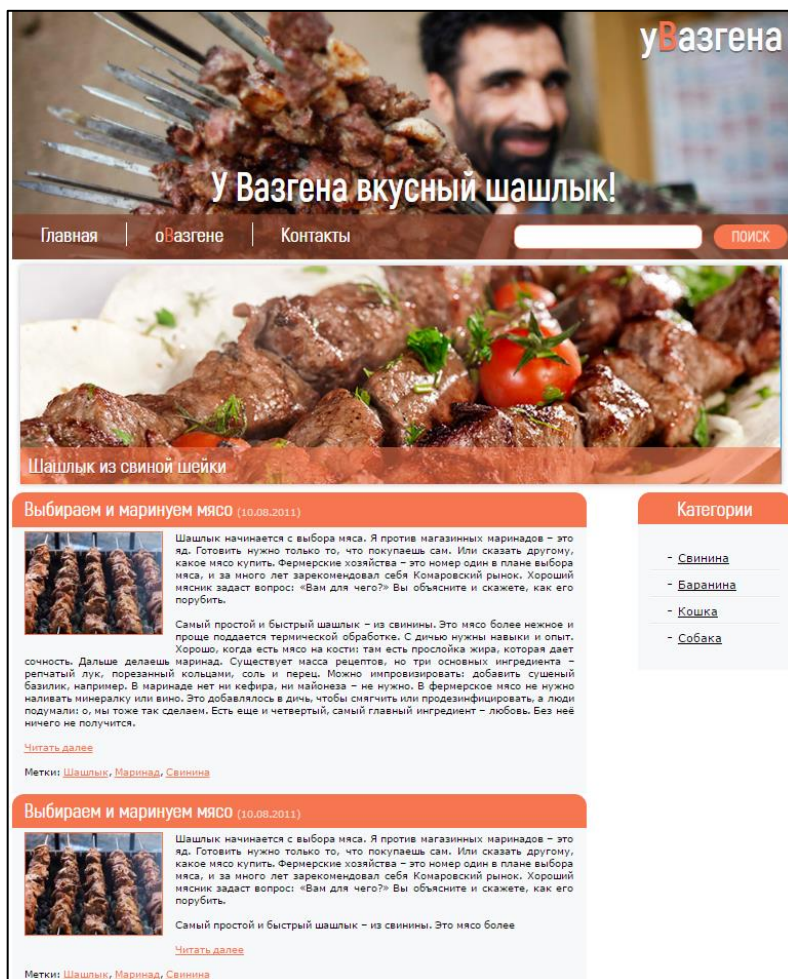


Рисунок 4 – «Шапка» и часть блока основного контента HTML-страницы с использованием стилей CSS

На рисунке 5 представлены «подвал» и оставшаяся часть блока основного контента HTML-страницы с использованием стилей CSS.

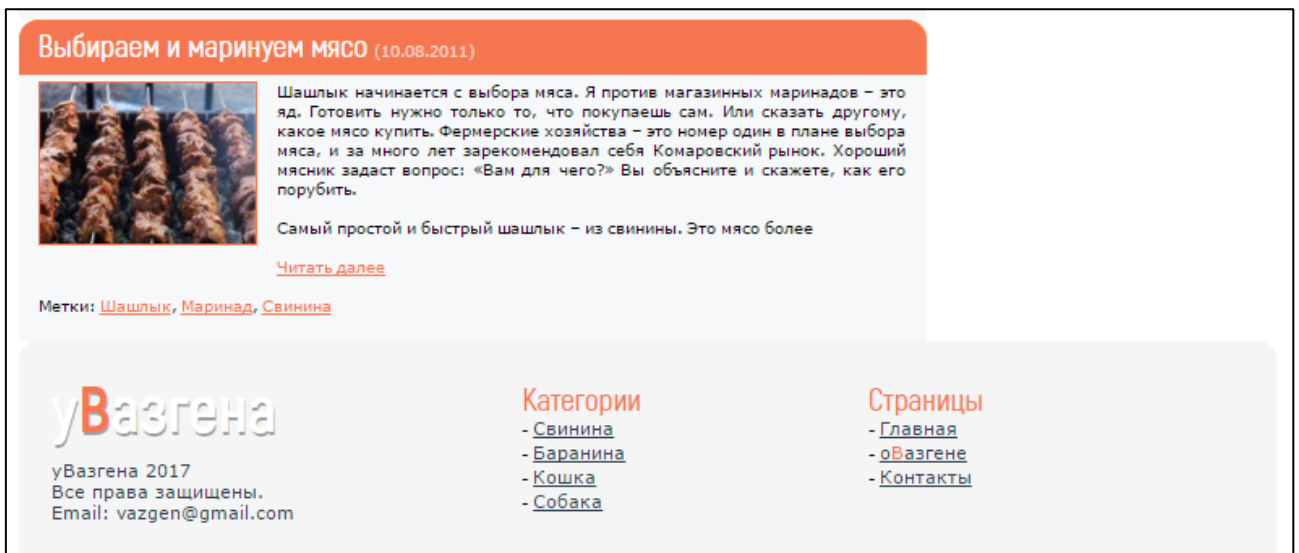


Рисунок 5 – «Подвал» и часть блока основного контента HTML-страницы с использованием стилей CSS.

В листинге 1 представлено содержимое файла index.html, в листинге 2 – содержимое файла style.css, в листинге 3 – содержимое файла orbit.css (стили CSS, используемые для ротатора картинок (слайдера)).

Листинг 1 – текст файла index.html

```

<!DOCTYPE html5>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <link rel="stylesheet" type="text/css" href="css/style.css" />
  <link rel="stylesheet" type="text/css" href="css/default0.css" media="all" />
  <style type="text/css">
    #featured { width: 940px; height: 300px; background: #009cff
      url('sliderImages/loading.gif') no-repeat center center;
overflow: hidden; }
  </style>
  <link rel="stylesheet" href="css/orbit.css">

  <script type="text/javascript" src="js/jquery-1.4.1.min.js"></script>
  <script type="text/javascript" src="js/jquery.orbit.min.js"></script>
  <script type="text/javascript">
    $(window).load(function() {
      $('#featured').orbit({
        'bullets': true,
        'timer' : true,
        'animation' : 'horizontal-slide'
      });
    });
  </script>

  <title>yВазгена</title>
  <style>
  .colortext {
    color: #f6774f;
  }
  </style>
</head>

<body>
  <div class = "main">
    <div class = "head">
      <h2>y Вазгена вкусный шашлык!</h2>
      <a href="/">y<span class="colortext">B</span>азгена</a>
    </div>
    <div class="content-main">
      <ul class = "menu">

```



```

                </p>
            </div>
            <div class="menu-foot">
                <h2>Категории</h2>
                <ul>
                    <li>
                        - <a href="#">Свинина</a>
                    </li>
                    <li>
                        - <a href="#">Баранина</a>
                    </li>
                    <li>
                        - <a href="#">Кошка</a>
                    </li>
                    <li>
                        - <a href="#">Собака</a>
                    </li>
                </ul>
            </div>
            <div class="menu-foot">
                <h2>Страницы</h2>
                <ul>
                    <li>
                        - <a href="#">Главная</a>
                    </li>
                    <li>
                        - <a href="#">о<span
class="colortext">в</span>азгене</a>
                    </li>
                    <li>
                        - <a href="#">Контакты</a>
                    </li>
                </ul>
            </div>
        </div>
    </div>
</body>
</html>

```

Листинг 2 – текст файла style.css

```

{
    padding: 0; margin: 0;
}
body{
    background: #fff url(../images/header1.jpg) center top no-repeat;
}
.main{
    width: 970px; margin: 0 auto;
}
.head{
    height: 260px; margin: 0 auto;
}
    .head h2{
        font-size: 61px;
        font-family: "Akrobat";
        color: rgb(255, 255, 255);
        font-weight: bold;
        line-height: 0.833;
        text-align: right;
        text-shadow: 0px 2px 0px rgba(0, 0, 0, 0.35);
        margin-top: 190px;
        margin-right: 120px;
    }
    .head a{
        font-size: 50px;
        font-family: "Akrobat";
        color: rgb(255, 255, 255);
        font-weight: bold;
        line-height: 0.833;
        text-align: right;
        text-shadow: 0px 2px 0px rgba(0, 0, 0, 0.35);
        margin-right: 30px;
        margin-top: -230px;
    }

```

```

        text-decoration: none;
        float: right;
    }
    .content-main{
/*background: #f7f7f7; /*border: 1px solid #ffffff*/; overflow: hidden;
margin-top: -190px;
    }
    .menu{
        list-style: none; /*background: #82402b;*/ overflow: hidden; /*opacity: 0.75;*/
    }
    .menu li{
float: left; background: url(../images/menu.jpg) left center no-repeat;
padding: 0px 35px;
    }
    .menu li:first-child{
        float: left; background: none; padding: 0px 35px;
    }
    .menu li a{
        font: 24px "Akrobat"; color: #ffffff; text-decoration: none;
    }
    .menu li a:hover{
        font: 24px "Akrobat"; color: #ffffff; text-decoration: underline;
    }

    .search-main{
        float: right; margin: -7px 25px 0 0;
    }
    .search-txt{
height: 30px; width: 230px; background: #ffffff; border: 1px solid #f6774f;
margin: 0 10px 0 0;
border-radius: 10px;
    }
    .search-btn{
        position: relative; top: 10px;
    }

    .content{
        width: 700px; float: left; margin: 0 0 0 0px;
    }
    .post-main{
        background: #f6f8f9; /*border: 1px solid #e7e2e3;*/ border-radius: 15px 15px 0 0;
        margin: 0 0 0px 0; padding: 0 0 5px 0;
    }
    .post-main h1 {
        font: normal 15px Verdana;color: ;background: #f6774f; border-radius: 15px 15px 0 0;
        padding: 5px 20px 8px 15px; margin: 0 0 5px 0;
    }
    .post-main h1 a{
        font: normal 24px "Akrobat";color: #ffffff; text-decoration: none;
    }
    .post-main h1 a:hover{
        font: normal 24px "Akrobat";color: #ffffff; text-decoration: underline;
    }
    .post-main h1 span{
        font: normal 12px Verdana; color: #f8d7cc;
    }
    .imgstyle{
        float: left; margin: 0 15px 15px 0; border: 1px solid #f6774f;
    }

    .post{
margin: 0 15px 15px 15px;
    }
    .post p{
        font: 12px Verdana; color: #000000; text-align: justify; margin: 0 0
15px 0;
    }
    .post p a{
        color: #f6774f; text-decoration: underline;
    }
    .post p a:hover{
        color: #f6774f; text-decoration: none;
    }

    .sidebar{
width: 207px; float: right; margin: 0 0 0 20px;
    }
    .vidget{

```

```

background: #f6f8f9; /*border: 1px solid #f8d7cc; */
border-radius: 15px 15px 0 0; margin: 0 20px 0 0;
}
        .vidget h2{
            font: 24px "Akrobat"; font-weight: 500; background:
#f6774f;
            color: #fff; text-align: center; padding: 5px 0;
            border-radius: 15px 15px 0 0;
        }
        .vidget ul{
            list-style: none; padding: 25px 15px;
        }
            .vidget ul li{
                border-bottom: 1px solid #eee9e9;
                border-top: 1px solid #ffffff;
                color: #000; font-family:
"Verdana";
                padding: 5px 0 5px 20px;
            }
            .vidget ul li:first-child{
                border-top: none;
            }
            .vidget ul li:last-child{
                border-bottom: none;
            }
            .vidget ul li a{
                font: 14px "Verdana"; color: #000;
            }
            .vidget ul li a:hover{
                font: 14px "Verdana"; color: #000;
            }
        }
        .footer{
            overflow: hidden; padding: 30px 25px; background: #f5f5f5;
            border-radius: 15px 15px 0 0;
        }
            .cpy{
                float: left; margin-right: 175px;
            }
                .cpy a{
                    font-size: 50px;
                    font-family: "Akrobat";
                    color: #fff;
                    font-weight: bold;
                    line-height: 0.833;
                    text-align: right;
                    text-shadow: 1px 2px 0px rgba(0, 0, 0, 0.35);
                    left: 750px;
                    top: 20px;
                    text-decoration: none;
                }
                .cpy p{
                    font: 14px Verdana; color: #2c3e50; margin: 20px 0 0 0;
                }
            }
            .menu-foot{
                float: left; margin-right: 175px;
            }
                .menu-foot h2{
                    font: 24px "Akrobat"; color: #f6774f;
                }
                .menu-foot ul{
                    list-style: none;
                }
                .menu-foot ul li a{
                    font: 14px "Verdana"; color: #2c3e50;
                }
                .menu-foot ul li a:hover{
                    font: 14px "Verdana"; color: #2c3e50;
                    text-decoration: none;
                }
            }

```

Листинг 3 – текст файла orbit.css

```

div.orbit { width: 1px; height: 1px; position: relative;
-moz-box-shadow: 0 0 10px rgba(0,0,0,.25);

```

```

    -webkit-box-shadow: 0 0 10px rgba(0,0,0,.25);
    overflow: hidden;
margin: 20px 0 10px 10px;}
div.orbit img { position: absolute; top: 0; left: 0; }
div.orbit a img { border: none; }

/* TIMER */
div.timer { width: 40px; height: 40px; overflow: hidden; position: absolute; top: 10px; right:10px;
opacity: .6; cursor: pointer; z-index: 1001; }
span.rotator { display: block; width: 40px; height: 40px; position: absolute; top: 0; left: -20px;
background: url('sliderImages/rotator-black.png') no-repeat; z-index: 3; }
span.mask { display: block; width: 20px; height: 40px; position: absolute; top: 0; right: 0; z-
index: 2; overflow: hidden; }
span.rotator.move { left: 0; }
span.mask.move { width: 40px; left: 0; background: url('sliderImages/timer-black.png') repeat 0px
0px; }
span.pause { display: block; width: 40px; height: 40px; position: absolute; top: 0; left: 0px;
background: url('sliderImages/pause-black.png') no-repeat; z-index: 4; opacity: 0; }
div.timer:hover span.pause,
span.pause.active,
div.timer:hover span.pause.active { opacity: 1; }

/* CAPTIONS */
div.caption { font: 24px "Akrobat"; background: #f6774f;
background: rgba(246,119,79,.6); width: 100%; z-index: 1000;
position: absolute; bottom:-100px; color: #fff; padding: 8px 0; }
div.caption span { padding: 0 10px; font-size: 24px; text-shadow: 0px 1px 0px rgba(0,0,0,.4);
margin: 0; }
.orbit-caption { display: none; }

/* DIRECTIONAL NAV */
div.orbit:hover div.slider-nav { display: block; }
div.slider-nav { display: none; }
div.slider-nav span { width: 33px; height: 33px; text-indent: -9999px; position: absolute; z-index:
1000; top: 43%; cursor: pointer; }
div.slider-nav span.right { background: url('sliderImages/right-arrow.png'); right: 10px; }
div.slider-nav span.left { background: url('sliderImages/left-arrow.png'); left: 10px; }

/* BULLET NAV */
.orbit-bullets { position: absolute; z-index: 1000; list-style: none; top: 10px; left: 7px; margin:
0; padding: 0; }
.orbit-bullets li { float: left; margin-left: 5px; cursor: pointer; color: #999; text-indent: -
9999px; background: url(sliderImages/bullets.png) no-repeat; 0 0; width: 7px; height: 7px; overflow:
hidden; }
.orbit-bullets li.active { color: #222; background-position: -7px 0; }

```

V. Знакомство со скриптовым языком PHP

Весь WordPress написан на PHP. Поэтому для того, чтобы создавать или редактировать темы, нужно знать основы языка. Мною были изучены некоторые основные возможности скриптового языка PHP: функции, числа, строки, переменные, массивы, условия, циклы и пр.

VI. Знакомство с CMS WordPress, изучение соответствующей документации

Прежде всего, WordPress — система управления содержимым сайта с открытым исходным кодом; написана на PHP; сервер базы данных — MySQL. Сфера применения — от блогов до достаточно сложных новостных ресурсов и интернет-магазинов. После теоретического ознакомления с особенностями WordPress (создание темы для выбранной системы управления сайтом, структура темы), он был установлен на OpenServer (перед установкой была создана пустая база данных в phpMyAdmin (веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД

MySQL) (рисунок 6), чтобы в дальнейшем можно было им пользоваться – создавать темы для WordPress локально.

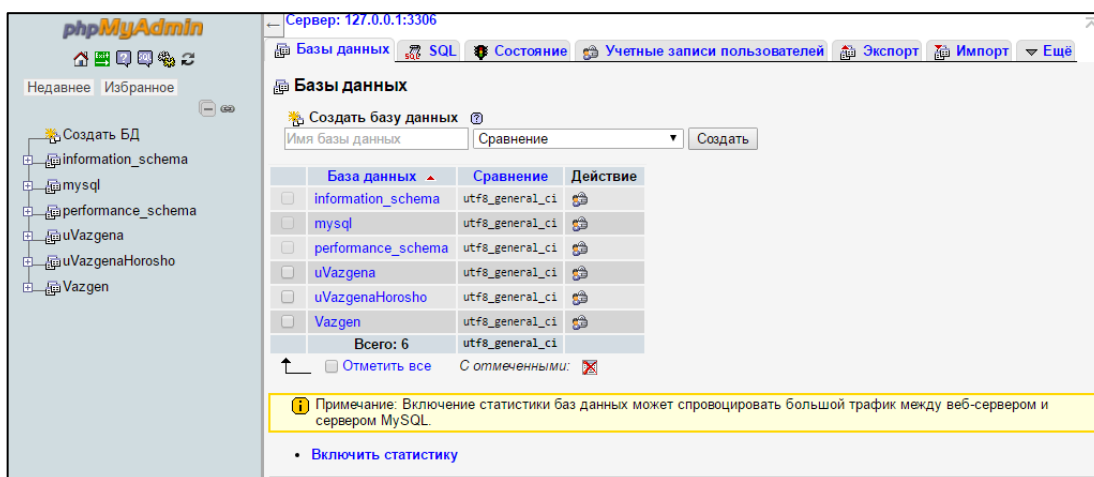


Рисунок 6 – Вкладка «Базы данных» в phpMyAdmin

После установки CMS WordPress и создания папки будущей темы (рисунок 7), были рассмотрены наиболее часто используемые функции WP (bloginfo(), have_posts(), the_permalink(), home_url() и пр.), знание которых является неотъемлемой частью при создании любой темы.

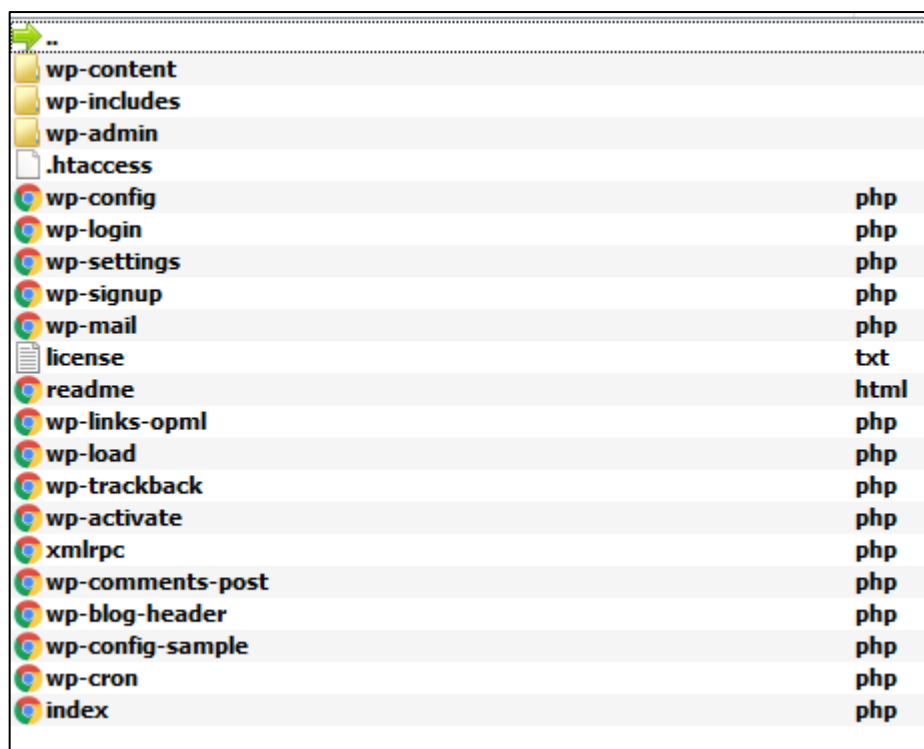


Рисунок 7 – Содержимое папки с темой WordPress

VII. Установка сайта на CMS WordPress

После краткого знакомства с функциями, уже имеющийся HTML шаблон был перенесён в папку с создаваемой темой. После чего он был разделён на логические части: header (шапка) (листинг 4), footer (подвал) (листинг 5), sidebar (боковая колонка) (листинг 6), которые позже были подключены к главному шаблону темы (index) (листинг 7).

Листинг 4 – текст файла header.php

```
<!DOCTYPE html5>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <link rel="stylesheet" type="text/css" href="<?php bloginfo('stylesheet_url'); ?>" />

  <style type="text/css">
    #featured { width: 940px; height: 300px; background: #009cff
      url('<?php bloginfo('template_url') ?>/images/loading.gif') no-
repeat center center; overflow: hidden; }
  </style>

  <script type="text/javascript" src="<?php bloginfo('template_url') ?>/js/jquery-
1.4.1.min.js"></script>
  <script type="text/javascript" src="<?php bloginfo('template_url')
?>/js/jquery.orbit.min.js"></script>

  <script type="text/javascript">
    $(window).load(function() {
      $('#featured').orbit({
        'bullets': true,
        'timer' : true,
        'animation' : 'horizontal-slide'
      });
    });
  </script>
<title>Отдыхай тело пой душа уВазгена <?php wp_title(); ?></title>
<style>
.colortext {
  color: #f6774f;
}
</style>
</head>
<body>

  <div class = "main">
    <div class = "head">
      <h2>У Вазгена вкусный шашлык!</h2>

      <a href="<?php echo home_url(); ?>">у<span
class="colortext">В</span>азгена</a>
    </div>
    <div class="content-main">
      <?php wp_nav_menu(array('theme_location' => 'menu', 'container' =>
false)); ?>

      <form class="search-main" action="<?php echo home_url('/'); ?>"
method="">
        <input class="search-txt" type="text" name="s" />
        <input class="search-btn" type="image" src="<?php
bloginfo('template_url') ?>/images/search-button.png" />
      </form>

      <?php $slider = new WP_Query(array('post_type' => 'slider',
'posts_per_page' => -1, 'order' => 'ASC')); ?>

      <?php if ($slider->have_posts()) : ?>

        <div id="featured">
          <ul id="cycle">

            <?php while ($slider->have_posts()): $slider->the_post(); ?>
              <li>
```



```

                <?php the_post_thumbnail('full'); ?>
            </li>
        <?php endwhile; ?>
    </ul>
    <div id="cyclePager">
    </div>
</div>
<?php else: ?>
    <p>
        Место под слайдер
    </p>
<?php endif; ?>
    <?php wp_reset_query(); ?>

```

Листинг 5 – текст файла footer.php

```

<div class="footer">
    <div class="copy">
        <a href="/">у<span class="colortext">В</span>азгена</a>
    <p>
        уВазгена 2017 <br />
        Все права защищены. <br />
        Email: <?php bloginfo( 'admin_email' ); ?>
    </p>
</div>
<?php if(!dynamic_sidebar('footer')) : ?>
    <div class="menu-foot">
        <h2>Категории</h2>
        <ul>
            <?php wp_list_categories(array('title_li' => '')); ?>
        </ul>
    </div>
<?php endif; ?>
<!--<div class="menu-foot">
    <h2>Категории</h2>
    <ul>
        <li>
            - <a href="#">Свинина</a>
        </li>
        <li>
            - <a href="#">Баранина</a>
        </li>
        <li>
            - <a href="#">Кошка</a>
        </li>
        <li>
            - <a href="#">Собака</a>
        </li>
    </ul>
</div-->
    <div class="menu-foot">
        <h2>Страницы</h2>
        <?php wp_nav_menu(array(
            'theme_location' => 'menu',
            'container' => false,
            'menu_class' => '',
            'before' => '- '
        )); ?>
    </div>
</div>
</div>
<?php wp_footer(); ?>
</body>
</html>

```

Листинг 6 – текст файла sidebar.php

```

<div class="sidebar">
    <?php if(!dynamic_sidebar('sidebar')) : ?>
        <div class="widget">
            <h2>Категории</h2>
            <ul>
                <?php wp_list_categories(array('title_li' => '')); ?>
            </ul>
        </div>
    </div>
<?php endif; ?>

```

```
</div>
```

Листинг 7 – текст файла index.php

```
<?php get_header(); ?>
    <div class="content">

<?php if (have_posts()) : while (have_posts()) : the_post(); ?>

    <div class="post-main">
        <h1><a href="<?php the_permalink(); ?>"><?php the_title(); ?></a>
        <span><?php the_time('j.m.Y') ?></span></h1>
        <div class="post">
            <?php the_post_thumbnail('full', 'class=imgstyle'); ?>
            <?php the_excerpt(); ?>
            <p>
                <a href="<?php the_permalink(); ?>">Читать далее</a>
            </p>
            <p>
                <?php the_tags(); ?>
            </p>
        </div>
    </div>
</div>

<?php endwhile; ?>

<div class="pagination">
    <?php
        //global $wp_query;

        $big = 999999999;

        echo paginate_links( array(
            'base' => str_replace($big, '%#%', esc_url(get_pagenum_link(
                $big
            ))),
            'format' => '?paged=%#%',
            'current' => max(1, get_query_var( 'paged' )),
            'total' => $wp_query->max_num_pages,
            'prev_text' => '&laquo;',
            'next_text' => '&raquo;';
        ) );
    ?>
</div>

<?php else: ?>
<?php endif; ?>
    </div>

<?php get_sidebar(); ?>
</div>
<?php get_footer(); ?>
```

В документации WordPress рекомендован вариант «безопасного подключения скриптов», поэтому я выбрал именно этот способ, безопасно подключив к теме скрипты и стили.

Далее в цикле WordPress был осуществлён вывод статей на главную страницу. После чего к теме была добавлена поддержка произвольных меню. Эти меню были созданы и выведены на сайт.

После чего к теме была добавлена поддержка виджетов, созданы области для виджетов в сайдбаре и футере. Добавлен виджет-аудиоплеер.

Так как кроме главной страницы с постами необходимо отображать другие страницы («Вазгене» и «Контакты») и страницы каждого отдельно взятого поста, были созданы два дополнительных шаблона:

- Шаблон отдельной статьи (поста) (single.php) (листинг 8);
- Шаблон страницы (page.php) (листинг 9).

На данном этапе тема почти готова. Остались лишь некоторые мелочи, которые в дальнейшем будут реализованы. В частности, нужно создать следующие шаблоны:

- Шаблон рубрик (категорий) (category.php) (листинг 10);
- Шаблон результатов поиска (search.php) (листинг 11);
- Шаблон 404-ой ошибки (404.php) (листинг 12);
- Шаблон комментариев (comments.php) (листинг 13).

Листинг 10 – текст файла category.php

```
<?php get_header(); ?>
    <div class="content">

<h2 class="page-title"><?php single_cat_title('Рубрика: '); ?></h2>

<?php if (have_posts()) : while (have_posts()) : the_post(); ?>

    <div class="post-main">
        <h1><a href="<?php the_permalink(); ?>"><?php the_title(); ?></a>
            <span><?php the_time('j.m.Y') ?></span></h1>
        <div class="post">
            <?php the_post_thumbnail('full', 'class=imgstyle'); ?>
            <?php the_excerpt(); ?>
            <p>
                <a href="<?php the_permalink(); ?>">Читать далее</a>
            </p>
            <p>
                <?php the_tags(); ?>
            </p>
        </div>
    </div>

<?php endwhile; ?>
<div class="pagination">
    <?php
        //global $wp_query;

        $big = 999999999;

        echo paginate_links( array(
            'base' => str_replace($big, '%#%', esc_url(get_pagenum_link(
                $big
            ))),
            'format' => '?paged=%#%',
            'current' => max(1, get_query_var( 'paged' )),
            'total' => $wp_query->max_num_pages,
            'prev_text' => '&laquo;',
            'next_text' => '&raquo;'
        ) );
    ?>
</div>
<?php else: ?>

<?php endif; ?>
    </div>
<?php get_sidebar(); ?>
    </div>
    <?php get_footer(); ?>
```

Листинг 11 – текст файла search.php

```
<?php get_header(); ?>
    <div class="content">

<h2 class="page-title">Результаты поиска: </h2>

<?php if (have_posts()) : while (have_posts()) : the_post(); ?>

    <div class="post-main">
        <h1><a href="<?php the_permalink(); ?>"><?php the_title(); ?></a>
            <span><?php the_time('j.m.Y') ?></span></h1>
```

```

        <div class="post">
            <?php the_post_thumbnail('full', 'class=imgstyle'); ?>
            <?php the_excerpt(); ?>
            <p>
                <a href="<?php the_permalink(); ?>">Читать далее</a>
            </p>
        </div>
    </div>

<?php endwhile; ?>
<div class="pagination">
    <?php
        //global $wp_query;

        $big = 999999999;

        echo paginate_links( array(
            'base' => str_replace($big, '%#%', esc_url(get_pagenum_link(
                $big
            ))),
            'format' => '?paged=%#%',
            'current' => max(1, get_query_var( 'paged' )),
            'total' => $wp_query->max_num_pages,
            'prev_text' => '&laquo;',
            'next_text' => '&raquo;'
        ) );
    ?>
</div>
<?php else: ?>
    <p>
        По запросу ничего не найдено...
    </p>
<?php endif; ?>

        </div>
<?php get_sidebar(); ?>
        </div>
    <?php get_footer(); ?>

```

Листинг 12 – текст файла 404.php

```

<?php get_header(); ?>

<div class="content">
    <h2 class="page-title">Ничего не найдено...</h2>
    
</div>
<?php get_sidebar(); ?>

<?php get_footer(); ?>

```

Листинг 13 – текст файла comments.php

```

<?php
if ( post_password_required() )
    return;
?>

<div id="comments" class="comments-area">

    <?php // You can start editing here -- including this comment! ?>

    <?php if ( have_comments() ) : ?>
        <h2 class="comments-title">
            <?php
                printf( _n( '1 комментарий к &ldquo;%2$s&rdquo;', '%1$s
комментариев(я) к &ldquo;%2$s&rdquo;', get_comments_number(), 'twentytwelve' ),
                    number_format_i18n( get_comments_number() ), '<span>' .
get_the_title() . '</span>' );
            ?>
        </h2>

        <ol class="commentlist">

```

```

        <?php wp_list_comments( array( 'callback' => 'twentytwelve_comment',
'style' => 'ol' ) ); ?>
        </ol><!-- .commentlist -->

        <?php if ( get_comment_pages_count() > 1 && get_option( 'page_comments' ) ) :
// are there comments to navigate through ?>
        <nav id="comment-nav-below" class="navigation" role="navigation">
            <h1 class="assistive-text section-heading"><?php _e( 'Навигация по
комментариям', 'twentytwelve' ); ?></h1>
            <div class="nav-previous"><?php previous_comments_link( __( '&larr;
Предыдущие комментарии', 'twentytwelve' ) ); ?></div>
            <div class="nav-next"><?php next_comments_link( __( 'Новые комментарии
&rarr;', 'twentytwelve' ) ); ?></div>
        </nav>
        <?php endif; // check for comment navigation ?>

        <?php

        if ( ! comments_open() && get_comments_number() ) : ?>
        <p class="nocomments"><?php _e( 'Комментирование закрыто' , 'twentytwelve' );
?></p>
        <?php endif; ?>

        <?php endif; // have_comments() ?>

        <?php comment_form(); ?>

</div>

```

Кроме данных шаблонов, нужно было реализовать ещё две вещи: постраничная навигация (пагинация) для материалов сайта и реализация управления слайдером (ротатором изображений) из «админки». Для решения последней задачи необходимо было познакомиться с пользовательскими типами записей и создать новый тип постов для сайта.

Все созданные функции (вместе с новыми зарегистрированными возможностями) представлены в листинге 14.

Листинг 14 – текст файла functions.php

```

<?php
/*
поддержка миниатюр
*/
add_theme_support('post-thumbnails');
/*
меню
*/
register_nav_menu('menu', 'Меню');
function load_style_script(){
    wp_enqueue_script('jquery-1', get_template_directory_uri()
'/js/jquery.orbit.min.js');
    wp_enqueue_script('jquery00', get_template_directory_uri() . '/js/jquery.orbit.js');
    wp_enqueue_script('init0000', get_template_directory_uri() . '/js/jquery-
1.4.1.min.js');
    wp_enqueue_style('style', get_template_directory_uri() . '/style.css');
}

add_action('wp_enqueue_scripts','load_style_script');
/*
виджеты в сайдбаре
*/
register_sidebar( array(
    'name' => 'Виджеты сайдбара',
    'id' => 'sidebar',
    'description' => 'Здесь размещайте виджеты сайдбара',
    'before_widget'=> '<div class="vidget">',
    'after_widget' =>'</div>',
    'before_title' => '<h2>',
    'after_title' => '</h2>'
) );

/*function audioShortcode() {
    return '[su_audio url="http://www.jplayer.org/audio/mp3/TSP-01-Cro_magnon_man.mp3?2"
width="200px" autoplay = "yes"]';
}

```

```

}
add_shortcode( 'AS', 'audioShortcode' );*/
/*
footer
*/
register_sidebar( array(
    'name' => 'Виджеты футера',
    'id' => 'footer',
    'description' => 'Здесь размещайте виджеты футера',
    'before_widget'=> '<div class="menu-foot">',
    'after_widget' =>'</div>',
    'before_title' => '<h2>',
    'after_title' => '</h2>'
) )
/*comments*/
function twentytwelve_comment( $comment, $args, $depth ) {
    $GLOBALS['comment'] = $comment;
    switch ( $comment->comment_type ) :
        case 'pingback' :
        case 'trackback' :
            // Display trackbacks differently than normal comments.
            ?>
            <li <?php comment_class(); ?> id="comment-<?php comment_ID(); ?>">
                <p><?php _e( 'Pingback:', 'twentytwelve' ); ?> <?php comment_author_link(); ?>
                <?php edit_comment_link( __( '(Edit)', 'twentytwelve' ), '<span class="edit-link">', '</span>' );
                ?></p>
                <?php
                    break;
                default :
                    // Proceed with normal comments.
                    global $post;
                    ?>
                    <li <?php comment_class(); ?> id="li-comment-<?php comment_ID(); ?>">
                        <article id="comment-<?php comment_ID(); ?>" class="comment">
                            <header class="comment-meta comment-author vcard">
                                <?php
                                    echo get_avatar( $comment, 44 );
                                    printf( '<cite><b class="fn">%1$s</b> %2$s</cite>',
                                        get_comment_author_link(),
                                        // If current post author is also comment author,
                                        // make it known visually.
                                        ( $comment->user_id === $post->post_author ) ?
                                        '<span>' . __( 'Post author', 'twentytwelve' ) . '</span>' : ''
                                    );
                                    printf(
                                        ' <a href="%1$s"><time
                                        datetime="%2$s">%3$s</time></a>',
                                        esc_url( get_comment_link( $comment->comment_ID )
                                    ),
                                        get_comment_time( 'c' ),
                                        /* translators: 1: date, 2: time */
                                        sprintf( __( '%1$s в %2$s', 'twentytwelve' ),
                                        get_comment_date(), get_comment_time() )
                                    );
                                ?>
                            </header><!-- .comment-meta -->
                            <?php if ( '0' == $comment->comment_approved ) : ?>
                                <p class="comment-awaiting-moderation"><?php _e( 'Ваш комментарий ожидает проверки', 'twentytwelve' ); ?></p>
                            <?php endif; ?>
                            <section class="comment-content comment">
                                <?php comment_text(); ?>
                                <?php edit_comment_link( __( 'Редактировать', 'twentytwelve' ),
                                ' <p class="edit-link">', '</p>' ); ?>
                            </section><!-- .comment-content -->
                            <div class="reply">
                                <?php comment_reply_link( array_merge( $args, array(
                                'reply_text' => __( 'Ответить', 'twentytwelve' ), 'after' => ' <span>&arr;</span>', 'depth' => $depth,
                                'max_depth' => $args['max_depth'] ) ) ); ?>
                            </div><!-- .reply -->
                        </article><!-- #comment-## -->
                    <?php
                        break;
                    endswitch; // end comment_type check
                }

```

```

/*slider*/
add_action('init', 'slider');
function slider() {
    register_post_type( 'slider', array(
        'public' => true,
        'supports' => array('title', 'thumbnail'),
        'labels' => array(
            'name' => 'Слайдер',
            'all_items' => 'Все слайды',
            'add_new' => 'Добавить новый',
            'add_new_item' => 'Добавление слайда'
        )
    ) );
}
?>

```

В ходе работы все стили были объединены в один файл и доработаны. Конечный файл стилей CSS объединяет в себе предыдущие файлы style.css, orbit.css, default.css.

Конечный внешний вид «админки» WordPress представлен на рисунке 8.

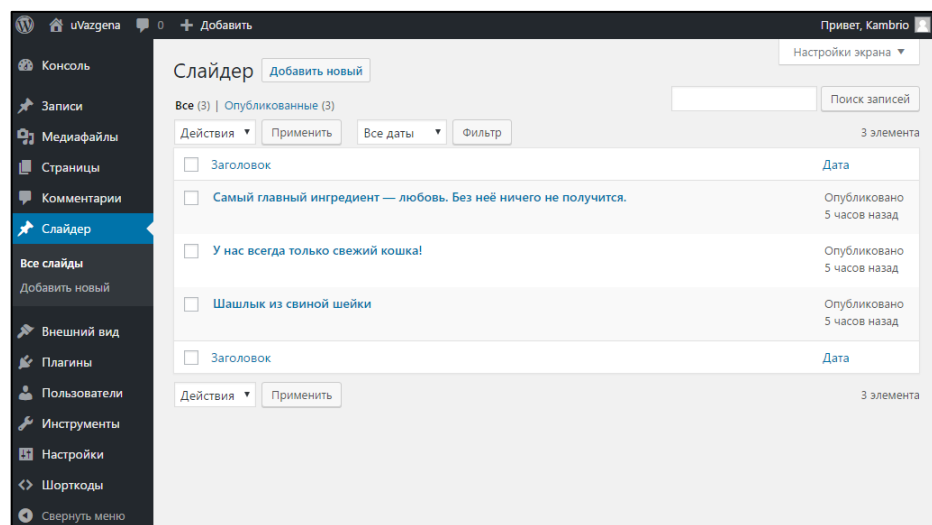


Рисунок 8 – Внешний вид WordPress
Фрагмент главной страницы представлен на рисунке 9.



Рисунок 9 – Фрагмент главной страницы (итоговый вариант)

На рисунке 10 представлен фрагмент страницы с текстом поста (статьи).

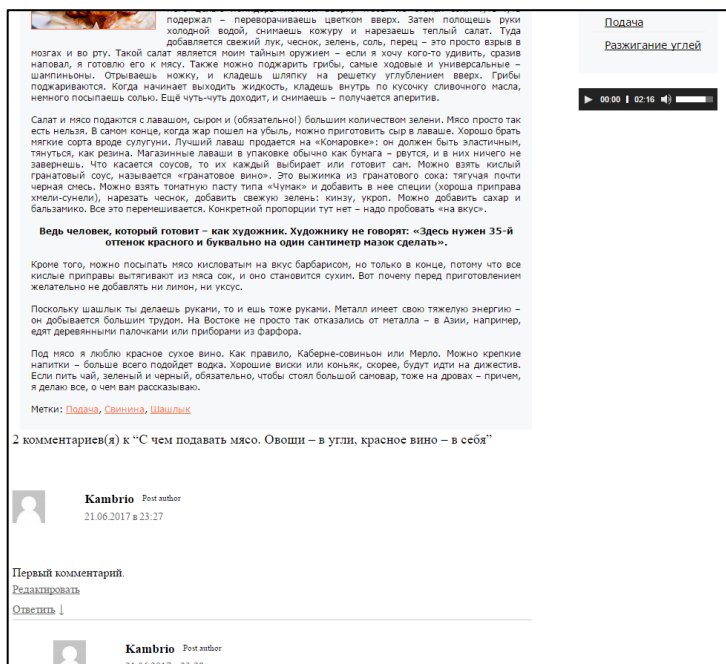


Рисунок 10 – Фрагмент страницы с текстом статьи

На рисунке 11 представлен фрагмент страницы «оВазгене».



Рисунок 11 – Фрагмент страницы «oВазгене»

Заключение

Актуальность прохождения учебной практики заключается в непосредственном знакомстве с процессом разработки сайта; приобретением практических навыков по специальности.

Первые несколько дней учебной практики были посвящены совершенствованию умения в разметке веб-страницы с помощью тегов HTML и приобретению навыков по оформлению веб-страницы с помощью стилей CSS.

Последующие дни практики были посвящены практическому использованию полученных за первые дни практики знаний, а именно: рисование макета сайта, его верстка и установка на CMS WordPress.

Установке сайта на CMS WordPress предшествовало ознакомление с WP, изучение его основных возможностей и функций, создание тем WordPress.

Учебная практика способствовала совершенствованию умения в разметке веб-страницы с помощью тегов HTML, приобретению навыков по оформлению веб-страницы с помощью таблицы стилей CSS, ознакомлению с установкой сайта на CMS WordPress.

Основная часть моей работы над заданием в период прохождения учебной практики составляла: рассмотрение и анализ материалов задания и дополнительных материалов, способствующих углублению знаний, необходимых для достижения поставленных целей, изучение тематической документации (учебные пособия, Интернет-ресурсы и документация по CMS WordPress), разработка веб-сайта и составление презентации, показывающей особенности его функционирования.

Последние дни учебной практики были посвящены реализации дополнительных функций, таких как постраничная навигация (пагинация) для материалов сайта и реализация управления слайдером (ротатором изображений) из «админки». Кроме того, была подготовлена презентация, демонстрирующая внешний вид и функционал созданного в течение учебной практики сайта.

Таким образом, учебная практика способствует реализации теоретических знаний, умений и навыков; развитию навыков в разметке веб-страницы с помощью тегов HTML, прикреплении таблицы стилей CSS, установке сайта на CMS WordPress. А также получению представления о практической деятельности по специальности.

Список использованных источников

1. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 3-е изд. — СПб.: Питер, 2015.
2. WordPress для начинающих / Дэрил Бартлетт ; [пер. с англ. М.А. Райтман]. – Москва : Издательство «Э», 2017. -208 с. – (Мировой компьютерный бестселлер).
3. WordPress Codex [Электронный ресурс]: офиц. сайт. - Режим доступа: <https://codex.wordpress.org/>, - свободный. – Загл. с экрана (дата обращения 22.06.2017).