

Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И  
РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Компьютерные системы и сети

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

Сайт-журнал

БГУИР КП I-40 01 01 312 ПЗ

Выполнил:

студент гр.351003

Левашко Н.А.

Проверил:

Третьяков Ф.И.

Минск, 2015

Задание  
по курсовому проектированию

Студенту Левашко Николаю Андреевичу

Тема работы Веб-приложение сайт-журнал

Срок сдачи 09.06.2015

Исходные данные документы формата .tpl, .php и .html

Содержание записки (перечень вопросов, которые подлежат разработке)

Введение. 1. Анализ прототипов литературных источников и  
формирование требований к проектируемому приложению. 2. Анализ  
требований к приложению и разработка функциональных требований.  
3.Проектирование, создание(конструирование) приложения.  
4. Тестированиеб проверка работоспособности и анализ полученных  
результатов. 5.Руководство по установке и использованию приложения.  
Заключение. Список использованной литературы. Приложения.

Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

Консультант по курсовой работе

Третьяков Ф.И.

Дата выдачи задания DD.MM.YYYY

Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и процентом от общего объёма работы):

раздел 1 к DD.MM.YYYY – 15 % готовности работы;

разделы 2, 3 к DD.MM.YYYY – 30 % готовности работы;

раздел 4 к DD.MM.YYYY – 60 % готовности работы;

раздел 5, 6 к DD.MM.YYYY – 90 % готовности работы;

оформление пояснительной записки и графического материала к \_\_\_\_\_  
DD.ММ.YYYY – 100 % готовности работы. \_\_\_\_\_

Защита курсового проекта с DD по DD июля YYYY г. \_\_\_\_\_

РУКОВОДИТЕЛЬ Третьяков Ф.И.

(подпись)

Задание принял к исполнению \_\_\_\_\_ Лёвошко Николай Андреевич \_\_\_\_\_

09.06.2015г.

(дата и подпись студента)

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ . . . . .	3
1 АНАЛИЗ ПРОТОТИПОВ, ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ И ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К ПРОЕКТИРУЕМОМУ ПРИЛОЖЕНИЮ . . . . .	5
1.1 Анализ литературных источников . . . . .	5
1.2 Анализ прототипов и программ аналогов . . . . .	6
1.3 Формирование требований к разрабатываемому приложению . . . . .	7
2 АНАЛИЗ ТРЕБОВАНИЙ К ПРИЛОЖЕНИЮ И РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ . . . . .	9
2.1 Описание функциональности приложения . . . . .	9
2.2 Спецификация функциональных требований . . . . .	10
3 ПРОЕКТИРОВАНИЕ, СОЗДАНИЕ (КОНСТРУИРОВАНИЕ) ПРИЛОЖЕНИЯ . . . . .	11
3.1 Описание используемых алгоритмов . . . . .	11
4 ТЕСТИРОВАНИЕ, ПРОВЕРКА РАБОТОСПОСОБНОСТИ И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ . . . . .	15
4.1 Проверка работоспособности . . . . .	15
4.2 Тестирование . . . . .	16
5 РУКОВОДСТВО ПО УСТАНОВКЕ И ИСПОЛЬЗОВАНИЮ ПРИЛОЖЕНИЯ . . . . .	17
ЗАКЛЮЧЕНИЕ . . . . .	20
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ . . . . .	21

ПРИЛОЖЕНИЕ А . . . . .	22
------------------------	----

## ВВЕДЕНИЕ

В современном мире высоко развиты сферы оказания услуг. Повсеместно можно увидеть предложения о оказании услуг в любой области. И интернет, как очень распространённая и большая часть информационного пространства, может помочь людям как найти предложения о услугах, так и опубликовать их. Самым широкоиспользуемым и применяемым методом являются различные веб-сайты.

С каждым днем численность интернет-аудитории растёт, тем самым расширяя возможности поиска потенциальных клиентов. Все большее число людей прибегают к поиску необходимых товаров и услуг в сети, и если Ваша продукция в отличие от конкурентов не представлена в интернете, Вы можете потерять значительную часть рынка.

Интернет во многом решил проблему коммуникации между отдаленными регионами. Имея собственный сайт, Вы можете с легкостью заявить о себе не только местным жителям, но и тем, кто проживает за тысячи километров от Вашего местонахождения.

Большинство людей предпочитают пользоваться услугами или покупать товары тех фирм, которые открыто рассказывают о своей деятельности. Веб-сайт – отличный способ продемонстрировать открытость и создать имидж надежной и серьезной компании.

Технологии современного веб-сайта позволяют создавать удобную систему обратной связи, которая помогает изучить мнения клиентов, выявить слабые места и оперативно среагировать на изменения на рынке. Кроме того, возможность доступной связи с производителем позволяет повысить лояльность к бренду.

Благодаря развитию методов разработки веб-сайтов, сейчас на Интернет-страничках можно размещать не только текстовое и

графическое описание услуг и товаров, но и публиковать видео-материалы, которые максимально полно продемонстрируют все преимущества предлагаемой продукции.

Интернет-сайт не требует присутствия на рабочем месте. И все желающие могут получить необходимую информацию о компании или заказать продукцию в любой момент времени.

Знакомство с продукцией на веб-сайте в отличие от надоевшей рекламы по радио и ТВ может быть гораздо эффективнее, поскольку так потенциальные клиенты будут получать интересующую их информацию (о Ваших товарах и услугах) по собственной воле.

На веб-сайте можно опубликовать всю возможную информацию о компании и продукции, контакты, схемы проезда, цены и прочее. Таким образом Вы избежите большого числа звонков и направите освободившееся время на развитие компании.

Представительский веб-сайт - еще один способ поиска и установления новых партнерских связей, которые дают немало возможностей для развития. При этом сфера поиска не будет ограничена географией. Нам повезло жить в эпоху развития информационных технологий и Интернета в частности. Всемирная сеть предлагает неисчерпаемые возможности для развития бизнеса любой сферы деятельности. Но в качестве основы всегда должен быть хороший веб-сайт, отвечающий целям и принципам организации и имеющий высокий показатель конверсии.

# **1 АНАЛИЗ ПРОТОТИПОВ, ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ И ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К ПРОЕКТИРУЕМОМУ ПРИЛОЖЕНИЮ**

## **1.1 Анализ литературных источников**

Сайт (от англ. website: web — «паутина, сеть» и site — «место», буквально «место, сегмент, часть в сети») — система электронных документов (файлов данных и кода) частного лица или организации в компьютерной сети под общим адресом (доменным именем или IP-адресом). Все сайты в совокупности составляют Всемирную паутину, где коммуникация (паутина) объединяет сегменты информации мирового сообщества в единое целое — базу данных и коммуникации планетарного масштаба. Для прямого доступа клиентов к сайтам на серверах был специально разработан протокол HTTP. Браузер — прикладное программное обеспечение для просмотра веб-страниц; содержания веб-документов, компьютерных файлов и их каталогов; управления веб-приложениями; а также для решения других задач. В глобальной сети браузеры используют для запроса, обработки, манипулирования и отображения содержания веб-сайтов. Многие современные браузеры также могут использоваться для обмена файлами с серверами ftp, а также для непосредственного просмотра содержания файлов многих графических форматов (gif, jpeg, png, svg), аудио-видео форматов (mp3, mpeg), текстовых форматов (pdf, djvu) и других файлов. Всемирная паутина — распределённая система, предоставляющая доступ к связанным между собой документам, расположенным на различных компьютерах, подключенных к Интернету. Для обозначения Всемирной паутины также используют слово веб (англ. web «паутина») и аббревиатуру WWW. Всемирную паутину образуют сотни миллионов веб-серверов. Большинство ресурсов всемирной паутины основаны на



технологии гипертекста. Гипертекстовые документы, размещаемые во Всемирной паутине, называются веб-страницами. Несколько веб-страниц, объединённых общей темой, дизайном, а также связанных между собой ссылками и обычно находящихся на одном и том же веб-сервере, называются веб-сайтом. Для загрузки и просмотра веб-страниц используются специальные программы — браузеры (англ. browser). Разработанное веб-приложение предназначается для размещения информации о предоставляемых услугах, а также размещения тематических статей. Разработаны функции для отображения вышеперечисленной информации. Веб-приложение не требует каких-либо входных данных. Обращение происходит путём ввода адреса сайта в адресную строку браузера. Выходными данными являются html-страницы, которые преобразуются браузером в удобный для чтения пользователю вид. Временные характеристики ограничиваются скоростью интернет соединения пользователя. Надёжность веб-приложения находится на высоком уровне, так как пользователь не может вносить какие-либо данные или как-либо влиять на работу сайта. Для разработки данного веб-приложения был выбран язык программирования php, так как он хорошо подходит для разработки шаблонизаторов. Также был выбран язык разметки HTML5, так как он широко используется для разработки веб-страниц сайтов.

## **1.2 Анализ прототипов и программ аналогов**

Сайт [texnoxors.ru](http://texnoxors.ru) — Сайт посвящён поставке подшипников, запорной арматуры, газового и теплоэнергетического оборудования. ТехноХОРС осуществляет поставки промышленной трубопроводной арматуры, применяемой в тепловой и атомной энергетике, химической, нефтяной, газовой и других отраслях промышленности. Сайт выполнен в стиле "визитки" с описанием товара и контактной информацией.



Сайт bkonsultant71.ru – Это сайт посвящён бухгалтерским услугам в Туле, оказываемых организациям и индивидуальным предпринимателям.

Этот сайт, так же как и предыдущий выполнен в виде "визитки". На главной странице отображён перечень оказываемых услуг и цены на них, а так же контактные данные.



### 1.3 Формирование требований к разрабатываемому приложению

На основе проведенного анализа требования к проектируемому программному средству являются следующими:

а) программа должна осуществлять отображение информации об

оказываемых услугах;

б) программа должна осуществлять изображение информации о контактах;

в) программа должна осуществлять размещение тематических статей.

г) программа должна предусматривать переход по страницам сайта.

д) предоставлять функцию доработки содержимого сайта.

Данная программа предназначена для личного использования.

Веб-приложение «Сайт-журнал» должно иметь ряд функций и процедур, организованных для выполнения поставленной задачи. Структура и описание таких функций и процедур будет подробно описано в последующих разделах.

Для реализации данной курсовой работы был выбран язык программирования PHP, позволяющий реализовать архитектуру приложения, а также язык разметки HTML5 и каскадная таблица стилей CSS3.

Данное приложение обладает удобным и понятным интерфейсом, который не требует каких-либо знаний программирования у пользователя и знаний в сфере информационных технологий в целом. То есть этой программой сможет воспользоваться любой пользователь, когда-либо работавший на компьютере.

Приложение требует минимальное количество оперативной памяти, занимает минимум места на жестком диске. Требуется наличия любого браузера, а также настроенного локального сервера Apache. Однако требуется наличие подключения к сети интернет.

## 2 АНАЛИЗ ТРЕБОВАНИЙ К ПРИЛОЖЕНИЮ И РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

### 2.1 Описание функциональности приложения

Для удобства доработки данного веб-приложения, был создан шаблонизатор на языке php. Концепция шаблонизатора состоит в том, чтобы разделять логику получения данных от логики отображения данных. Шаблонизатор — программное обеспечение, позволяющее использовать html-шаблоны для генерации конечных html-страниц.

Основная цель использования шаблонизаторов — это отделение представления данных от исполняемого кода. Часто это необходимо для обеспечения возможности параллельной работы программиста и дизайнера-верстальщика. Использование шаблонизаторов часто улучшает читаемость кода и внесение изменений во внешний вид, когда проект целиком выполняет один человек. Сайт был разработан в минималистичном стиле, однако он содержит всю необходимую информацию о предоставляемых услугах. Таким образом реализованы следующие пункты:

- Сайт не требует пристального внимания, обслуживания и постоянного обновления контента.
- Достаточно простой сайт, предоставляет только самую важную информацию.
- Возможно модернизировать сайт в будущем

Данное веб-приложение содержит информацию о контактах, разнообразные тематические статьи, информацию об оказываемых услугах, а также стоимость этих услуг. Всё это отображается в виде слайдера, а так же в виде текста с изображениями на страницах сайта.

Пользователь, после отправки запроса на сайт, сможет увидеть главную страницу сайта, и затем перейти на нужную страницу.

## **2.2 Спецификация функциональных требований**

1. Пользователю должен быть предоставлен выбор, на какую страницу сделать переход.
2. Пользователь должен иметь возможность получить информацию об оказываемых услугах.
3. Пользователь должен иметь возможность получить информацию о контактах.
4. Пользователь должен иметь возможность чтения различных тематических статей.

## 3 ПРОЕКТИРОВАНИЕ, СОЗДАНИЕ (КОНСТРУИРОВАНИЕ) ПРИЛОЖЕНИЯ

### 3.1 Описание используемых алгоритмов

#### Шаблонизатор

Концепция шаблонизатора состоит в том, чтобы разделять логику получения данных от логики отображения данных.

Шаблонизатор находит при помощи регулярных выражений и записывает в массив строк, содержащий название tpl файлов, в файле страницы, затем извлекает содержимое этих файлов и подставляет их в готовую для вывода веб-страницу. Шаблонизатор реализован при помощи двух методов, таких как `getplaceholders`, который находит названия tpl файлов и записывает их в массив и `output`, который располагает содержимое этих файлов в структуру страницы. В файлах самих страниц, на который поступает запрос содержится лишь подключение шаблонизатора, создание экземпляра класса и вывод содержимого готовой страницы.

Реализация шаблонизатора:

```
<?php
class Template
{
private $file;

public function __construct($file)
{
$this->file = $file;
}

public function output()
```

```

{

$output = file_get_contents($this->file);
$changed = true;
while ($changed)
{
    $changed = false;
    $placeholders = $this->get_placeholders();
    foreach ($placeholders as $key => $value)
    {
        $tag_to_replace = $value;
        $output_copy = $output;
        $template_name = 'templates/' . str_replace("{include(\"", "", $value)
        $template_name = str_replace("\")}", "", $template_name) . '.tpl';
        $output = str_replace($tag_to_replace, file_get_contents($template_name), $output);
        if ($output !== $output_copy)
        {
            $changed = true;
        }
    }
}

return $output;
}

private function get_placeholders()
{
    $template = file_get_contents($this->file);
    $pattern = '/{include\(.+?\)}'/;

```

```

$matches = array();
preg_match_all($pattern, $template, $matches);

return $matches[0];
}

}

```

Сами tpl файлы содержат html-код, при помощи которого конструируется будущий вид страницы. В файлах страниц содержится подключение таблицы стилей, для более удобного и красочного отображения страницы, а так же файлы javascript, в которых реализован слайдер, используемый на двух страницах сайта, был использован слайдер easyslider. Структура сайта состоит из заголовка, плавающего меню, блока с информационной частью. В заголовке находится название страницы и ссылка на главную страницу сайта. В плавающем меню расположены ссылки на остальные страницы сайта. В информационном блоке содержится информация, зависящая от открытой пользователем страницы. Реализация одного tpl файла:

```

<div id="menu">
<ul>
<li><a class="links" href="contacts.php">Контакты</a></li>
<li><a class="links" href="price.php">Услуги и цены</a></li>
<li><a class="links" href="topics.php">Статьи</a></li>
</ul>
</div>
<div id="content">

```



```

<div id="container">

<div id="sliderheader">
<h1>Качественная проводка</h1>
</div>
<div id="slidercontent">
<div id="slider">
<ul>
<li><a href=""><
<li><a href=""><
<li><a href=""><
<li><a href=""><
<li><a href=""><
</ul>
</div>
</div>

</div>

</div>

```

Реализация файла страницы:

```

<?php
include("template.php");
$template = new Template("templates/page-template.html");
echo $template->output();

```

## 4 ТЕСТИРОВАНИЕ, ПРОВЕРКА РАБОТОСПОСОБНОСТИ И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

### 4.1 Проверка работоспособности

На изображениях ниже приведены доказательства корректной работы веб-приложения.

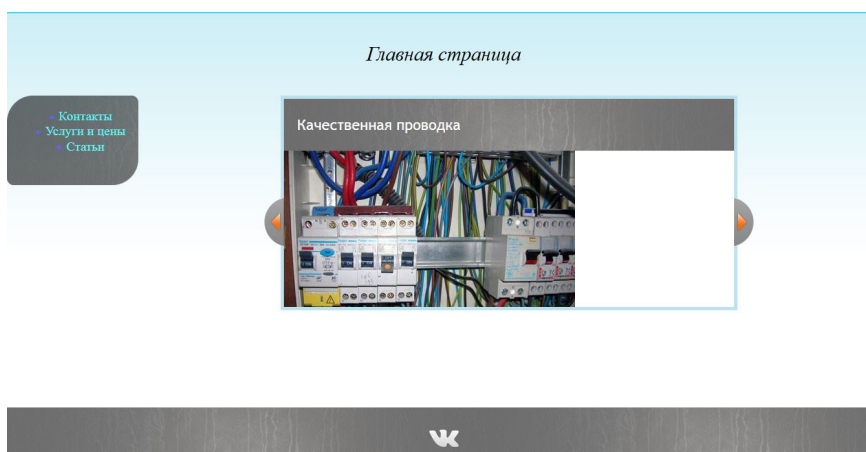


Рис.1 Работа главной страницы сайта. Изображено меню со ссылками на другие страницы, слайдер и информационные элементы страницы.

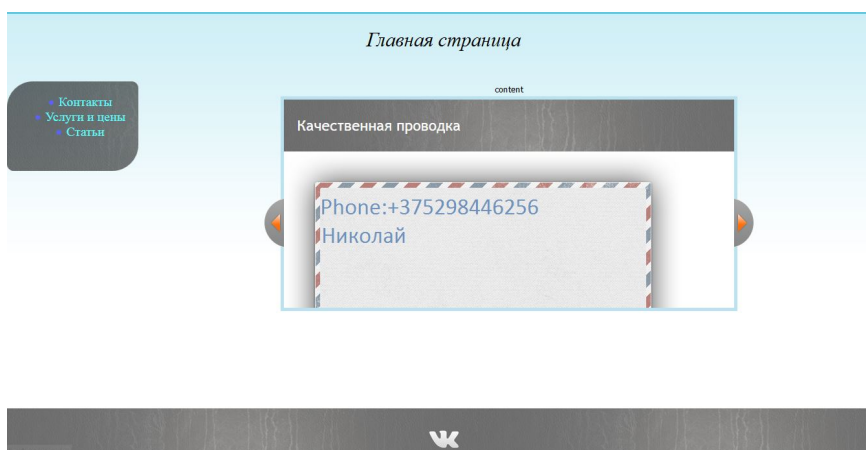


Рис.2 Работа контактной страницы сайта. Изображено меню со ссылками на другие страницы, слайдер и контактный данные.

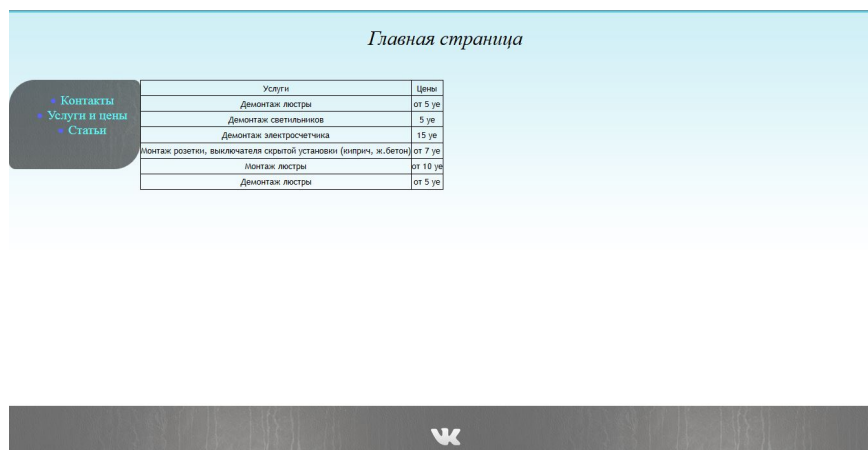


Рис.3 Работа страницы сайта, содержащий услуги и цены. Изображено меню со ссылками на другие страницы, предоставляемые услуги и их стоимость.



Рис.4 Работа страницы сайта, содержащей статьи. Изображено меню со ссылками на другие страницы, тематические статьи.

## 4.2 Тестирование

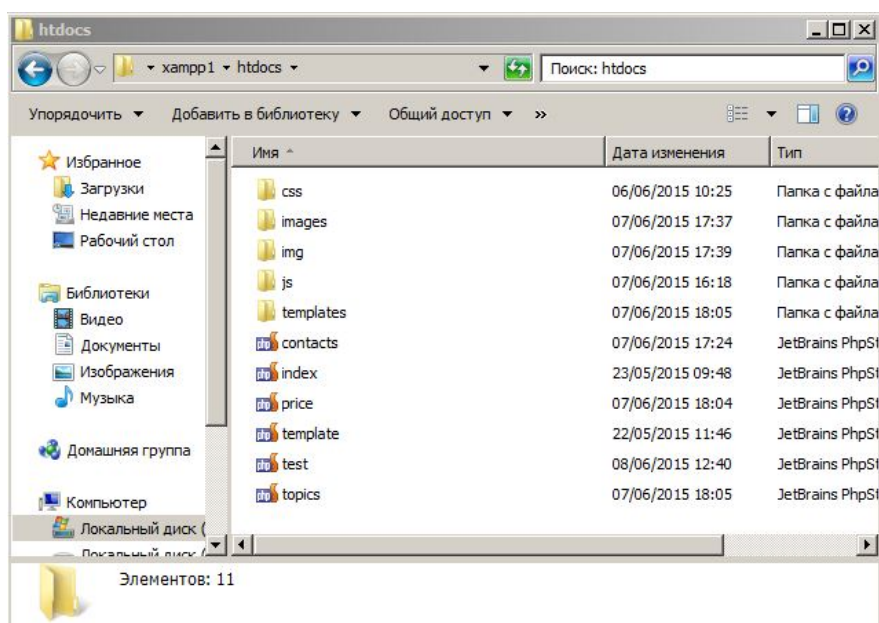
Тестирование программы происходило при помощи PHPUnit. Были протестированы 2 метода.

## 5 РУКОВОДСТВО ПО УСТАНОВКЕ И ИСПОЛЬЗОВАНИЮ ПРИЛОЖЕНИЯ

Прежде чем запускать приложение необходимо настроить локальный сервер Apache. Подробную инструкцию по установке и настройке Apache вы можете увидеть здесь:

<http://www.webbcare.org/lectures/web-servers/Lecture15.pdf>.

Можно также воспользоваться уже настроенными пакетами, такими как xampp (<https://www.apachefriends.org/ru/index.html>) или denver (<http://www.denwer.ru/>). Файлы, при распаковке, следует поместить в папку htdocs (все скрипты, запускаемые на локальной машине по умолчанию, находятся в данной папке) в уже настроенном Apache.

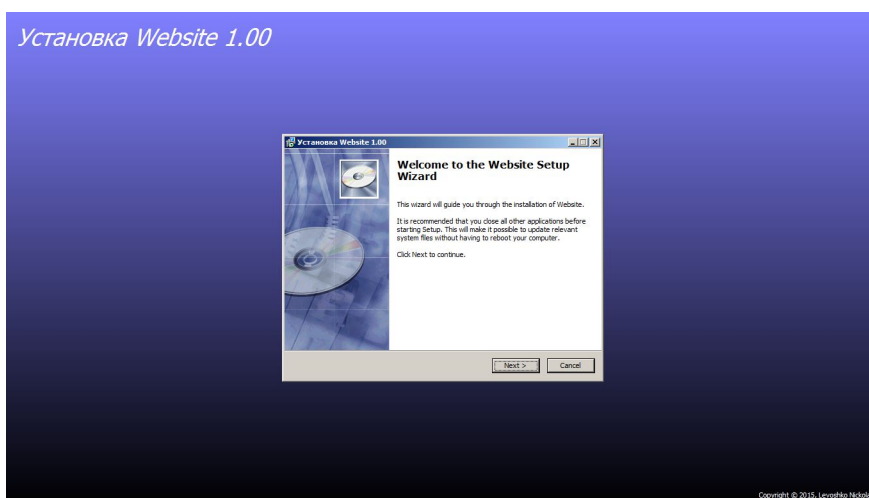


Далее запускаем браузер. В адресной строке необходимо прописать путь к файлу: `http://localhost/htdocs/(название страницы, либо оставить пустым)`. Приложение готово к использованию.

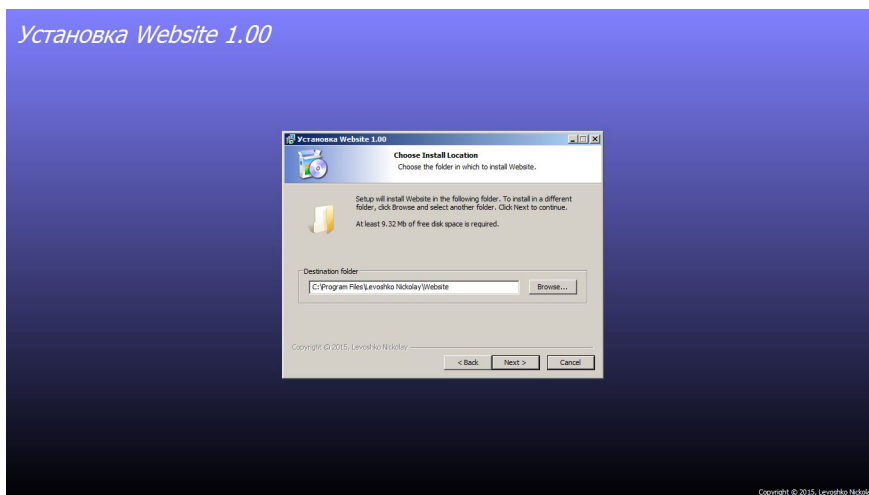
Так же можно воспользоваться установочным файлом SetupWebsite.exe. Файл был создан при помощи программы Smart Install Maker. После запуска программы необходимо следовать инструкции установки.



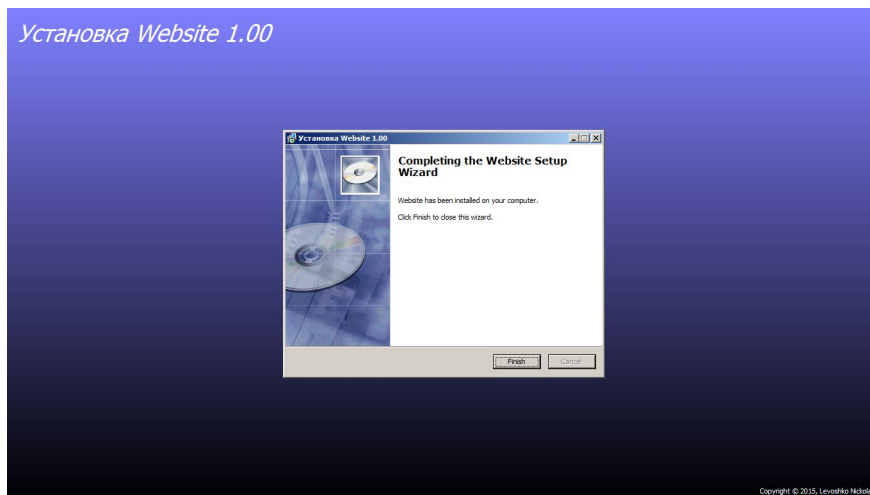
Первый шаг установки- выбор языка установки.



Второй шаг установки- подтверждение.



Третий шаг установки - выбор директории установки.



Последний шаг установки.  
Приложение готово к использованию.

## ЗАКЛЮЧЕНИЕ

В результате выполнения данного курсового проекта было разработано веб-приложение «Сайт-журнал», позволяющее получать информацию об оказываемых услугах, контактах, были выполнены условия, требуемые в условии курсового проекта, работает корректно и без сбоев.

На данный момент веб-приложение имеет следующий функционал:

- отображение информации об услугах и их ценах; -отображение информации о контактах; -отображение тематических статей;

Работа была разделена на этапы, такие как анализ прототипов, литературных источников и постановка требований к проектируемому приложению, разработка алгоритма и блок-схемы его работы, конструирование, отладка и тестирование. После последовательного выполнения вышеперечисленных этапов разработки, было получено исправно работающее приложение.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- [1] Html and css for real world , Авторы: Alexis Goldstein, Louis Lazaris, Estelle Weyl.
- [2] ГОСТ 19.701–90. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения. – Введ. 1992–01–01. – М. : Изд-во стандартов, 1991.
- [3] JavaScript.ru [Электронный ресурс]. – События : <http://htmlbook.ru>.
- [4] Habrahabr.ru [Электронный ресурс]. – React : <http://habrahabr.ru/>



## ПРИЛОЖЕНИЕ А(обязательное)

Код шаблонизатора

```
<?php
class Template
{
private $file;

public function __construct($file)
{
$this->file = $file;
}

public function output()
{

$output = file_get_contents($this->file);
$changed = true;
while ($changed)
{
$changed = false;
$placeholders = $this->get_placeholders();
foreach ($placeholders as $key => $value)
{
$tag_to_replace = $value;
$output_copy = $output;
$template_name = 'templates/' . str_replace("{include(\"", "", $value)
$template_name = str_replace("\")}", "", $template_name) . '.tpl';
$output = str_replace($tag_to_replace, file_get_contents($template_name
```

```

if ($output !== $output_copy)
{
    $changed = true;
}
}
}
return $output;
}

private function get_placeholders()
{
    $template = file_get_contents($this->file);
    $pattern = '/{include\(.+?\)}'/;

    $matches = array();
    preg_match_all($pattern, $template, $matches);

    return $matches[0];
}

}

```

Код главной страницы

```

<?php
include("template.php");
$template = new Template("templates/page-template.html");
echo $template->output();

```

Код страницы контактов

```
<?php
include("template.php");
$template = new Template("templates/page-template-contacts.html");
echo $template->output();
```

Код страницы цен и услуг

```
<?php
include("template.php");
$template = new Template("templates/page-template-price.html");
echo $template->output();
```

Код страницы статей

```
<?php
include("template.php");
$template = new Template("templates/page-template-topics.html");
echo $template->output();
```

Код html-документа

```
<?php
<!DOCTYPE html>
<html >
<head>
    <title>{include("title")}</title>
<link href="{stylesheet}" rel="stylesheet" />
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="css/index.css" />
<link rel="stylesheet" type="text/css" href="css/normalize.css"/>
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/easySlider1.7.js"></script>
```

```

<link href="css/screen.css" rel="stylesheet" type="text/css" media="sc
<script type="text/javascript" src="js/sliderstart.js"></script>
</head>
<body>
    {include("header")}
    {include("content-index")}
{include("footer")}
</body>

</html>

```

Код footer

```

<footer>
<p><a href="https://vk.com/id220129264"> Главная страница</a></p>
</header>

```

Код содержания главной страницы

```

<div id="menu">
<ul>
<li><a class="links" href="contacts.php">Контакты</a></li>
<li><a class="links" href="price.php">Услуги и цены</a></li>
<li><a class="links" href="topics.php">Статьи</a></li>

```

```

</ul>
</div>
<div id="content">

<div id="container">

<div id="sliderheader">
<h1>Качественная проводка</h1>
</div>
<div id="slidercontent">
<div id="slider">
<ul>
<li><a href=""><
<li><a href=""><
<li><a href=""><
<li><a href=""><
<li><a href=""><
</ul>
</div>
</div>

</div>

</div>

```

Код содержания страницы контактов

```

<div id="menu">
<ul>
<li><a class="links" href="contacts.php">Контакты</a></li>

```

```

<li><a class="links" href="price.php">Услуги и цены</a></li>
<li><a class="links" href="topics.php">Статьи</a></li>
</ul>
</div>
<div id="content">content

<div id="container">

<div id="sliderheader">
<h1>Качественная проводка</h1>
</div>
<div id="slidercontent">
<div id="slider">
<ul>
<li><a href=""></a></li>
<li><a href=""></a></li>
<li><a href=""></a></li>
</ul>
</div>
</div>

</div>

</div>

```

Код easyslider

```

/*
 * Easy Slider 1.7 - jQuery plugin
 * written by Alen Grakalic

```

```

* http://cssglobe.com/post/4004/easy-slider-15-the-easiest-jquery-plu
*
* Copyright (c) 2009 Alen Grakalic (http://cssglobe.com)
* Dual licensed under the MIT (MIT-LICENSE.txt)
* and GPL (GPL-LICENSE.txt) licenses.
*
* Built for jQuery library
* http://jquery.com
*
*/

/*
* markup example for $("#slider").easySlider();
*
* <div id="slider">
* <ul>
* <li></li>
* <li></li>
* <li></li>
* <li></li>
* <li></li>
* </ul>
* </div>
*
*/

(function($) {

$.fn.easySlider = function(options){

```

```

// default configuration properties
var defaults = {
  prevId:  'prevBtn',
  prevText: 'Previous',
  nextId:  'nextBtn',
  nextText: 'Next',
  controlsShow: true,
  controlsBefore: '',
  controlsAfter: '',
  controlsFade: true,
  firstId:  'firstBtn',
  firstText: 'First',
  firstShow: false,
  lastId:  'lastBtn',
  lastText: 'Last',
  lastShow: false,
  vertical: false,
  speed: 800,
  auto: false,
  pause: 2000,
  continuous: false,
  numeric: false,
  numericId: 'controls'
};

var options = $.extend(defaults, options);

this.each(function() {

```



```

var obj = $(this);
var s = $("li", obj).length;
var w = $("li", obj).width();
var h = $("li", obj).height();
var clickable = true;
obj.width(w);
obj.height(h);
obj.css("overflow","hidden");
var ts = s-1;
var t = 0;
$("ul", obj).css('width',s*w);

if(options.continuous){
$("ul", obj).prepend($("ul li:last-child", obj).clone().css("margin-le
$("ul", obj).append($("ul li:nth-child(2)", obj).clone()));
$("ul", obj).css('width',(s+1)*w);
};

if(!options.vertical) $("li", obj).css('float','left');

if(options.controlsShow){
var html = options.controlsBefore;
if(options.numeric){
html += '<ol id="'+ options.numericId +'"></ol>';
} else {
if(options.firstShow) html += '<span id="'+ options.firstId +'"><a href
html += ' <span id="'+ options.prevId +'"><a href=\"javascript:void(0)
html += ' <span id="'+ options.nextId +'"><a href=\"javascript:void(0)
if(options.lastShow) html += ' <span id="'+ options.lastId +'"><a href

```

```

};

html += options.controlsAfter;
$(obj).after(html);
};

if(options.numeric){
for(var i=0;i<s;i++){
$(document.createElement("li"))
.attr('id',options.numericId + (i+1))
.html('<a rel='+ i +' href=\"javascript:void(0);\">' + (i+1) +'</a>')
.appendTo($("#"+ options.numericId))
.click(function(){
animate($("#a",$(this)).attr('rel'),true);
});
};
} else {
$("#a",$("#"+options.nextId).click(function(){
animate("next",true);
});
$("#a",$("#"+options.prevId).click(function(){
animate("prev",true);
});
$("#a",$("#"+options.firstId).click(function(){
animate("first",true);
});
$("#a",$("#"+options.lastId).click(function(){
animate("last",true);
});
};

```

```

};

function setCurrent(i){
i = parseInt(i)+1;
$("li", "#" + options.numericId).removeClass("current");
$("li#" + options.numericId + i).addClass("current");
};

function adjust(){
if(t>ts) t=0;
if(t<0) t=ts;
if(!options.vertical) {
$("ul",obj).css("margin-left",(t*w*-1));
} else {
$("ul",obj).css("margin-left",(t*h*-1));
}
clickable = true;
if(options.numeric) setCurrent(t);
};

function animate(dir,clicked){
if (clickable){
clickable = false;
var ot = t;
switch(dir){
case "next":
t = (ot>=ts) ? (options.continuous ? t+1 : ts) : t+1;
break;
case "prev":

```

```

t = (t<=0) ? (options.continuous ? t-1 : 0) : t-1;
break;
case "first":
t = 0;
break;
case "last":
t = ts;
break;
default:
t = dir;
break;
};
var diff = Math.abs(ot-t);
var speed = diff*options.speed;
if(!options.vertical) {
p = (t*w*-1);
$("ul",obj).animate(
{ marginLeft: p },
{ queue:false, duration:speed, complete:adjust }
);
} else {
p = (t*h*-1);
$("ul",obj).animate(
{ marginTop: p },
{ queue:false, duration:speed, complete:adjust }
);
};

if(!options.continuous && options.controlsFade){

```

```

if(t==ts){
$("a","#"+options.nextId).hide();
$("a","#"+options.lastId).hide();
} else {
$("a","#"+options.nextId).show();
$("a","#"+options.lastId).show();
};

if(t==0){
$("a","#"+options.prevId).hide();
$("a","#"+options.firstId).hide();
} else {
$("a","#"+options.prevId).show();
$("a","#"+options.firstId).show();
};
};

if(clicked) clearTimeout(timeout);
if(options.auto && dir=="next" && !clicked){
timeout = setTimeout(function(){
animate("next",false);
},diff*options.speed+options.pause);
};

};

// init
var timeout;
if(options.auto){

```

```

timeout = setTimeout(function(){
    animate("next",false);
},options.pause);
};

if(options.numeric) setCurrent(0);

if(!options.continuous && options.controlsFade){
    $("a","#"+options.prevId).hide();
    $("a","#"+options.firstId).hide();
};

});

};

})(jQuery);

```

Код запуска слайдера

```

$(document).ready(function(){
    $("#slider").easySlider({
        auto: true,
        continuous: true
    });
});

```

## Код стиля главной страницы

```
@font-face {
    font-family: ariston_normal;
    src: url(ariston_normal.ttf);
}

header{
    font-size: 23pt;
    font-style: italic;
    text-align: center;
    color: black;
    font-family: ariston_normal;
}

body {
    background-image: url("../img/fonst.jpg");
    background-size: cover;
}

#menu {
    opacity: 0.6;
    height:20%;
    background-image: url("../img/fonm.jpg");
    position: fixed;
    width: 15%;
    color:blue;
    border-radius: 40px 10px;
    font-size: 15pt;
    font-family: ariston_normal;
}

#content {
```

```
position: absolute;
margin-left: 15%;
color: black;
height: 100%;
width: 85%;
}

footer {
opacity: 0.6;
background-image: url("../img/fonm.jpg");
position: fixed;
color: white;
bottom: 0;
width: 100%;
height: 11%;
font-size: 15pt;
font-family: ariston_normal;
}

.links {
text-decoration: none;
color: aqua;
}

.mainlinks {
text-decoration: none;
color: black;
}

.icons {
width: 55px;
height: 55px;
margin-left: 0%;
```



```
margin-top: 0%;  
}
```

## Код стиля слайдера

```
body {  
background:#fff url(../images/bg_body.gif) repeat-x;  
font:80% Trebuchet MS, Arial, Helvetica, Sans-Serif;  
color:#333;  
line-height:180%;  
margin:0;  
padding:0;  
text-align:center;  
}  
h1{  
font-size:180%;  
font-weight:normal;  
margin:0;  
padding:0 20px;  
}  
h2{  
font-size:160%;  
font-weight:normal;  
}  
h3{  
font-size:140%;  
font-weight:normal;  
}  
img{border:none;}  
pre{
```

```

display:block;
font:12px "Courier New", Courier, monospace;
padding:10px;
border:1px solid #bae2f0;
background:#e3f4f9;
margin:.5em 0;
width:674px;
}

```

```

/* image replacement */
.graphic, #prevBtn, #nextBtn, #slider1prev, #slider1next{
    margin:0;
    padding:0;
    display:block;
    overflow:hidden;
    text-indent:-8000px;
}

```

```

/* // image replacement */

```

```

#container{
margin:0 auto;
position:relative;
text-align:left;
width:696px;
background:#fff;
margin-bottom:2em;
border: 5px solid #bae2f0;
}

#sliderheader{

```

```
opacity: 0.6;
height:80px;
line-height:80px;
background:#5DC9E1;
color: White;
background-image: url("../img/fonm.jpg");
```

```
}
```

```
#slidercontent{
position:relative;
}
```

```
/* Easy Slider */
```

```
#slider ul, #slider li,
#slider2 ul, #slider2 li{
margin:0;
padding:0;
list-style:none;
}
```

```
#slider2{margin-top:1em;}
```

```
#slider li, #slider2 li{
```

```
/*
```

```
define width and height of list item (slide)
```

```
entire slider area will adjust according to the parameters provided he
```

```
*/
```

```
width:696px;
```

```
height:241px;
```

```

overflow:hidden;
}

#prevBtn, #nextBtn,
#slider1next, #slider1prev{
display:block;
width:30px;
height:77px;
position:absolute;
left:-30px;
top:71px;
z-index:1000;
}

#nextBtn, #slider1next{
left:696px;
}

#prevBtn a, #nextBtn a,
#slider1next a, #slider1prev a{
display:block;
position:relative;
width:30px;
height:77px;
background:url(../images/btn_prev.gif) no-repeat 0 0;
}

#nextBtn a, #slider1next a{
background:url(../images/btn_next.gif) no-repeat 0 0;
}

/* numeric controls */

```

```

ol#controls{
margin:1em 0;
padding:0;
height:28px;
}
ol#controls li{
margin:0 10px 0 0;
padding:0;
float:left;
list-style:none;
height:28px;
line-height:28px;
}
ol#controls li a{
float:left;
height:28px;
line-height:28px;
border:1px solid #ccc;
background:#DAF3F8;
color:#555;
padding:0 10px;
text-decoration:none;
}
ol#controls li.current a{
background:#5DC9E1;
color:#fff;
}
ol#controls li a:focus, #prevBtn a:focus, #nextBtn a:focus{outline:none

```

```

/* // Easy Slider */
.sm_slider {
    height: 270px;
    margin: 25px 25px 0px;
    position: relative;
    width: 960px;
}

.sm_slider ul {
    list-style: none;
    margin: 0px;
    padding: 0px;
}

.sm_slide {
    color: #3d3d3d;
    display: none;
    font-size: 100px;
    height: 270px;
    left: 0;
    position: absolute;
    text-align: center;
    top: 0;
    visibility: hidden;
    width: 100%;
}

.sm_slide.active {
    display: block;
    visibility: visible;
}

```

```
.sm_slide img {  
    max-width: 100%;  
}
```

```
.sm_prev,  
.sm_next {  
    width: 25px;  
    height: 25px;  
    margin: -20px 0 0 0;  
    cursor: pointer;  
    display: block;  
    font-style: normal;  
    opacity: .3;  
    position: absolute;  
    top: 50%;  
    z-index: 999;  
    border-style: solid;  
    border-color: #4e4e4e;  
    border-width: 0px;  
    -webkit-transform: rotate(-45deg);  
    -moz-transform: rotate(-45deg);  
    -ms-transform: rotate(-45deg);  
    -o-transform: rotate(-45deg);  
    transform: rotate(-45deg);  
}
```

```
.sm_prev {  
    border-left-width: 3px;  
    border-top-width: 3px;
```

```

    left: 10px;
}

.sm_prev:hover {
    opacity: 1;
}

.sm_next {
    border-right-width: 3px;
    border-bottom-width: 3px;
    right: 10px;
}

.sm_next:hover {
    opacity: 1;
}

.sm_nav {
    bottom: 0px;
    clear: both;
    display: inline-block;
    overflow: hidden;
    padding: 5px 0px;
    position: absolute;
    right: 0px;
    z-index: 999;
}

.sm_nav-item {
    cursor: pointer;
    float: left;

```



```

height: 14px;
padding: 4px;
width: 14px;
z-index: 999;
list-style-type: none;
font-size: 11px;
color: #ccc;
}

.sm_nav-item i {
display: block;
width: 6px;
height: 6px;
position: relative;
top: 5px;
left: 4px;
-webkit-border-radius: 6px;
-moz-border-radius: 6px;
-ms-border-radius: 6px;
-o-border-radius: 6px;
border-radius: 6px;
background-color: #7f7f7f;
}

.sm_nav-item.active {
cursor: default;
color: #7e7e7e;
}

.sm_nav-item.active:hover {
cursor: default;
color: #7e7e7e;
}

```

```
}  
.sm_nav-item.active i {  
  background-color: #bfbfbf;  
}  
.sm_nav-item:hover i {  
  background: #7f7f7f;  
}  
.sm_nav-item.active:hover i {  
  background-color: #bfbfbf;  
}
```