МИНОБРНАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования «Вологодский государственный университет»

Кафедра «Автоматики и вычислительной техники»

Дисциплина «Технологии интернет программирования»

**КУРСОВАЯ РАБОТА**

Выполнил студент:

Михеев Владислав Алексеевич

Группа: ЭПИ-41

Вологда 2018 г.

**Содержание**

[Введение 3](#_Toc446863567)

[1.Технологии веб-систем 4](#_Toc446863568)

[1.1.Серверные языки 4](#_Toc446863569)

[1.2.HTML 5](#_Toc446863570)

[1.3.JavaScript 6](#_Toc446863571)

[1.4.CSS 7](#_Toc446863572)

[2.Фреймворк 8](#_Toc446863573)

[2.1.Bootstrap 8](#_Toc446863574)

[3.Разработка сайта с помощью Bootstrap. 11](#_Toc446863575)

[Заключение. 16](#_Toc446863576)

[Список литературы 17](#_Toc446863577)

# **Введение**

Веб-программирование — раздел программирования, ориентированный на разработку динамических Internet-приложений. Языки веб-программирования — это соответственно языки, которые в основном предназначены для работы с интернет технологиями. К примеру, для написания сайта нужно минимум знание одного из языков программирования, а в среднем - 2-3, чтобы в итоге вышел хороший динамичный сайт. Развитие Веб-программирования не стоит на месте, языки написания совершенствуются. Будет рассмотрены некоторые из них, появление самого первого и сущность современного, актуального в настоящее время, языка веб программирования.

World Wide Web - глобальная компьютерная сеть на сегодняшний день содержит миллионы сайтов, на которых размещена всевозможная информация. Люди получают доступ к этой информации посредством использования технологии Internet. Для навигации в WWW используются специальные программы - Web-браузеры, которые существенно облегчают путешествие по бескрайним просторам WWW. Вся информация в Web-браузере отображается в виде Web-страниц, которые являются основным элементом байтов WWW. Пользователю приятно посещать те Web-страницы, которые имеют стильное оформление, не отягощены чрезмерно графикой и анимацией, быстро загружаются и правильно отображаются в окне Web-браузера.

Создать Web-страницу непросто, но наверно каждый человек хотел бы попробовать себя в роли дизайнера. И я, в данном случае, не являюсь исключением и выполнил практическую работу по разработке собственного Web - сайта.

# **Технологии веб-систем**

Все технологии, применяемые при создании веб-систем, делятся на два основных класса: исполняемые на клиенте средствами обозревателя Интернет (HTML, CSS, JavaScript, Flash, ActiveX и т.п.) и исполняемые на сервере средствами веб-сервера (SSI, PHP, ASP, Perl, Python и т.п.) и связанных с ним систем (MySQL, PostgreSQL, MSSQL и т.п.). Приложения, исполняемые на сервере, практически ничем не ограничены по сложности: могут выполнять любые преобразования информации и затем формировать поток данных, который может быть визуализирован пользователю обозревателем Интернет. Форматы данных, которые могут быть обработаны на клиенте, ограничены достаточно узким набором технологий, стандартов и определенными рамками, что позволяет унифицировать рабочее место пользователя и не требовать от него установки какого-либо дополнительного программного обеспечения кроме обозревателя Интернет.

Языки веб-программирования -- это языки, которые в основном предназначены для работы с веб-технологиями. Языки веб-программирования можно условно разделить на две пересекающиеся группы: клиентские и серверные.

# **Серверные языки**

Когда пользователь дает запрос на какую-либо страницу (переходит на неё по ссылке или вводит адрес в адресной строке своего браузера), то вызванная страница сначала обрабатывается на сервере, то есть выполняются все программы, связанные со страницей, и только потом возвращается к посетителю по сети в виде файла. Этот файл может иметь расширения: HTML, PHP, ASP, ASPX, Perl, SSI, XML, DHTML, XHTML.

# **HTML**

Web-страницы могут существовать в любом формате, но в качестве стандарта принят Hyper Text Markup Language - язык разметки гипертекстов, предназначенный для создания форматированного текста, насыщенного изображениями, звуком, анимацией, видеоклипами и гипертекстовыми ссылками на другие документы, разбросанные как по всему Web-пространству, так и находящиеся на этом же сервере или являющиеся составной частью этого же Веб-проекта.

Можно работать на Web без знания языка HTML, поскольку тексты HTML могут создаваться разными специальными редакторами и конвертерами. Но писать непосредственно на HTML нетрудно. Возможно, это даже легче, чем изучать HTML-редактор или конвертер, которые часто ограничены в своих возможностях, содержат ошибки или проводят плохой HTML код, который не работает на разных платформах.

Язык HTML существует в нескольких вариантах и продолжает развиваться, но конструкции HTML скорее всего будут использоваться и в дальнейшем. Изучая HTML и познавая его глубже, создавая документ в начале изучения HTML и расширяя его насколько это возможно, мы имеем возможность создавать Web-страницы, которые могут быть просмотрены многими браузерами Web, как сейчас, так и в будущем. Это не исключает возможности использования других методов, например, метод расширенных возможностей, который предоставляется Netscape Navigator, Internet Explorer или некоторыми другими программами.

Работа по HTML - это способ усвоить особенности создания документов в стандартизированном языке, используя расширения, только если это действительно необходимо.

# **Javacript**

JavaScript является языком сценариев (скриптов), который применяют в основном для создания на Web-страницах интерактивных элементов. Его можно использовать для построения меню, проверки правильности заполнения форм, смены изображений или для чего-то еще, что можно сделать на Web-странице. Если взглянуть на Google Maps или службу GMail компании Google, то можно понять, на что способен сегодня язык JavaScript.

Так как JavaScript является в настоящее время единственным языком сценариев, который поддерживают все основные браузеры Web (Internet Explorer, Firefox, Netscape, Safari,Opera, Camino и т.д.), то он используется очень широко.

Код JavaScript обычно выполняется Web-браузером клиента, и в этом случае он называется сценарием на стороне клиента. Но код JavaScript можно выполнять также на Web-сервере для формирования документов HTML, воплощая тем самым сценарий на стороне сервера. Хотя использование JavaScript обычно ограничивается сценариями на стороне клиента, он является также очень мощным серверным языком.

При создании кода JavaScript требуется фактически только текстовый редактор и Web-браузер. Знание HTML и CSS будет играть определенно положительную роль, и если вы захотите использовать навыки JavaScript на Web-сайте, то понадобится также Web-сайт. Если у вас уже есть Web-сайт, то отлично! Если нет, то существует множество бесплатных серверов, которые можно использовать для размещения своих страниц. Что касается текстового редактора, то в Windows имеется редактор NotePad. Хотя этого будет достаточно для редактирования JavaScript, HTML и CSS, более мощный редактор, такой, например, как EditPlus или другой, может оказаться более удобным.

# **CSS**

**CSS** — [формальный язык](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) описания внешнего вида документа, написанного с использованием[языка разметки](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8).

Преимущественно используется как средство описания, оформления внешнего вида [веб-страниц](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0), написанных с помощью [языков разметки](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8) [HTML](https://ru.wikipedia.org/wiki/HTML) и [XHTML](https://ru.wikipedia.org/wiki/XHTML), но может также применяться к любым [XML-документам](https://ru.wikipedia.org/wiki/XML), например, к [SVG](https://ru.wikipedia.org/wiki/SVG) или [XUL](https://ru.wikipedia.org/wiki/XUL).

CSS используется создателями [веб-страниц](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0) для задания [цветов](https://ru.wikipedia.org/wiki/%D0%A6%D0%B2%D0%B5%D1%82), [шрифтов](https://ru.wikipedia.org/wiki/%D0%A8%D1%80%D0%B8%D1%84%D1%82), расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы (которое производится с помощью [HTML](https://ru.wikipedia.org/wiki/HTML) или других [языков разметки](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8)) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью [формального языка](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или [методах](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4) вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими [шрифт Брайля](https://ru.wikipedia.org/wiki/%D0%A8%D1%80%D0%B8%D1%84%D1%82_%D0%91%D1%80%D0%B0%D0%B9%D0%BB%D1%8F).

# **Фреймворк**

Фреймворк ([неологизм](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D0%B7%D0%BC) от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *framework* — каркас, структура) — [программная платформа](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B0), определяющая структуру программной системы; [программное обеспечение](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), облегчающее разработку и объединение разных компонентов большого программного проекта. Употребляется также слово «каркас», а некоторые авторы используют его в качестве основного, в том числе не базируясь вообще на англоязычном аналоге. Можно также говорить о [каркасном подходе](https://ru.wikiversity.org/wiki/%D0%9A%D0%B0%D1%80%D0%BA%D0%B0%D1%81_%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D1%8B)[[3]](https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA#cite_note-gorbunov-3) как о подходе к построению программ, где любая конфигурация программы строится из двух частей: первая, постоянная часть — каркас, не меняющийся от конфигурации к конфигурации и несущий в себе гнёзда, в которых размещается вторая, переменная часть — сменные модули (или точки расширения)

# **Bootstrap**

Bootstrap (также известен как Twitter Bootstrap) — свободный набор инструментов для создания [сайтов](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82) и [веб-приложений](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5). Включает в себя [HTML](https://ru.wikipedia.org/wiki/HTML) и[CSS](https://ru.wikipedia.org/wiki/CSS) шаблоны оформления для [типографики](https://ru.wikipedia.org/wiki/%D0%A2%D0%B8%D0%BF%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0" \o "Типографика), веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая [JavaScript](https://ru.wikipedia.org/wiki/JavaScript" \o "JavaScript)-расширения.

Bootstrap использует самые современные наработки в области CSS и HTML, поэтому необходимо быть внимательным при поддержке старых браузеров.

Этот фреймворк начал разрабатываться в качестве внутренней библиотеки компании [Twitter](https://ru.wikipedia.org/wiki/Twitter" \o "Twitter) под названием Twitter Blueprint. После нескольких месяцев разраработки он был открыт под названием Bootstrap [19 августа](https://ru.wikipedia.org/wiki/19_%D0%B0%D0%B2%D0%B3%D1%83%D1%81%D1%82%D0%B0) [2011 года](https://ru.wikipedia.org/wiki/2011_%D0%B3%D0%BE%D0%B4).

Основными нововведениями второй версии, появившейся 31 января 2012 года, стали 12 колоночная сетка и поддержка [адаптивности](https://ru.wikipedia.org/wiki/%D0%90%D0%B4%D0%B0%D0%BF%D1%82%D0%B8%D0%B2%D0%BD%D1%8B%D0%B9_%D0%B2%D0%B5%D0%B1-%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD).

Третья версия была выпущена 19 августа 2013 года. В ней адаптивность получила дальнейшее развитие, был осуществлён переход к концепции [mobile first](https://ru.wikipedia.org/w/index.php?title=Mobile_first&action=edit&redlink=1" \o "Mobile first (страница отсутствует)), оптимизации прежде всего под мобильные устройства. Дизайн по умолчанию стал [плоским](https://ru.wikipedia.org/wiki/Flat_design).

Работа над четвёртой версией начата 29 октября 2014 года. Альфа версия вышла 19 августа 2015 года.

Основные преимущества Bootstrap 3:

* Экономия времени — Bootstrap позволяет сэкономить время и усилия, используя шаблоны дизайна и классы, и сконцентрироваться на других разработках;
* Высокая скорость — динамичные макеты Bootstrap масштабируются на разные устройства и разрешения экрана без каких-либо изменений в разметке;
* Гармоничный дизайн — все компоненты платформы Bootstrap используют единый стиль и шаблоны с помощью центральной библиотеки. Дизайн и макеты веб-страниц согласуются друг с другом;
* Простота в использовании — платформа проста в использовании, пользователь с базовыми знаниями HTML и CSS может начать разработку с Twitter Bootstrap;
* Совместимость с браузерами — Twitter Bootstrap совместим с Mozilla Firefox, Yandex Browser, Google Chrome, Safari, Internet Explorer и Opera;
* Открытое программное обеспечение — особенность Twitter Bootstrap, которая предполагает удобство использования, посредством открытости исходных кодов и бесплатной загрузки.

29 октября 2014 года Марк Отто объявил, что Bootstrap 4 находится в разработке. 6 сентября 2016 года Марк приостановил работу над Bootstrap 3, чтобы высвободить время для работы над Bootstrap 4. На текущий момент было внесено более 4000 изменений к базовому коду Bootstrap 4. Первая стабильная версия вышла 19 января 2018 года.

Bootstrap 4 - это почти полностью переписанный Bootstrap 3. Перечень самых значительных изменений:

* Веб-шрифты по умолчанию (Helvetica Neue, Helvetica, Arial) интегрированы в Bootstrap 4 и заменены «стаком (набором) нативных шрифтов» для оптимального рендеринга текста на любой машине или устройстве под любой ОС.
* Переход от использования Less к Sass.
* Не поддерживаются IE8, IE9 и iOS 6.
* Добавлена поддержка Flexbox, а затем отключена поддержка non flexbox.
* Переключение от использования пикселей к корневым ems.
* Увеличенный глобальный размер шрифта с 14px до 16px.
* Удалены панели, миниатюры и компоненты в пользу карт.
* Удален шрифт значков Glyphicons.
* Удалены компоненты пейджера.
* Переписаны почти все компоненты, плагины jQuery и документация.

Основные инструменты Bootstrap:

* Сетки — заранее заданные размеры колонок, которые можно сразу же использовать, например ширина колонки 140px относится к классу .span2 (.col-md-2 в третьей версии фреймворка), который можно использовать в CSS описании документа.
* Шаблоны — Фиксированный или резиновый шаблон документа.
* Типографика — Описания шрифтов, определение некоторых классов для шрифтов, таких как код, цитаты и т. п.
* Медиа — Представляет некоторое управление изображениями и Видео.
* Таблицы — Средства оформления таблиц, вплоть до добавления функциональности сортировки.
* Формы — Классы для оформления форм и некоторых событий происходящих с ними.
* Навигация — Классы оформления для Табов, Вкладок, Страничности, Меню и Тулбара.
* Алерты — Оформление диалоговых окон, Подсказок и Всплывающих окон.

# **Разработка сайта с помощью Bootstrap.**

Преимущество использования CSS фреймворков заключается в том, что верстальщику не нужно думать о многих нюансах верстки, которые за него уже продумали создатели фреймворков. К таким нюансам относятся кроссбраузерность, поддержка различных разрешений экранов и многое другое. Верстальщик лишь указывает, что, как и когда нужно показать, остальное фреймворк делает сам. Данный подход может сильно ускорить вёрстку сайта. К преимуществам Bootstrap относится и его популярность. Это означает, что другому верстальщику будет проще поддерживать ваш код.  
 Недостатком использования фреймворков является тот факт, что странице придется целиком «нести» за собой лишние стили фреймворка, даже если она использует лишь их малую часть. Фреймворк является отличным инструментом для прототипирования и создания страниц, для которых дизайн вторичен, например, страницы администрирования. Если же у вас есть очень специфический дизайн, то сверстать его с помощью фреймворка может оказаться сложнее, чем нативными средствами. Тем не менее, и это возможно.  
 Первым шагом давайте создадим простую структуру файлов для нашего проекта.

* 1. Создаем папку с названием проекта.
  2. Затем нужно [скачать сам Bootstrap](https://github.com/twbs/bootstrap/releases/download/v3.1.0/bootstrap-3.1.0-dist.zip) и скопировать содержимое архива в папку проекта.
  3. Создаем в папке проекта html файлы с подключением к bootstrap.

Bootstrap оснащен HTML, CSS и JS для всех видов работы, они все перечислены в категориях, которые вы можете найти в верхней части страницы [по документации Bootstrap](http://mybootstrap.ru/).

Список компонентов

Все вместе **компоненты** и **плагины Javascript** содержат следующие элементы интерфейса:

* Группы кнопок
* Выпадающие списки кнопок
* Навигационные вкладки, кнопки и списки
* Панель навигации
* Ярлыки
* Бейджи
* Заголовки страниц и элемент hero
* Миниатюры
* Сообщения
* Индикаторы процесса
* Модальные элементы
* Выпадающие списки
* Всплывающие подсказки
* Информационные блоки
* Элемент «Гармошка»
* Элемент «Карусель»
* Опережающий ввод с клавиатуры

Согласно документации, не сложно разобраться в Bootstrap и сделать свою страницу.

После подключения Bоotstrap к проекту возможно использовать параметры, которые заложены в css файлах для верстки страниц.

Описание работы с Bootstrap:

В коде, предоставленном ниже можно увидеть, каким образом реализована навигация и основная структура сайта.

Код на HTML:

<div id="wrapper">

<!-- Sidebar -->

<div id="sidebar-wrapper">

<ul class="sidebar-nav">

<li class="sidebar-brand">

<a href="index.html" class="main-name">

<img src="img/logo.svg" alt="">

</a>

</li>

<li>

<a href="graphics.html">Графики</a>

</li>

<li>

<a href="technology.html">Используемые технологии</a>

</li>

<li>

<a href="contact.html">Обратная связь</a>

</li>

</ul>

</div>

<!-- /#sidebar-wrapper -->

<!-- Page Content -->

<div id="page-content-wrapper">

<div class="container-fluid">

<div class="nav-icon" id="menu-toggle">

<span></span>

<span></span>

<span></span>

<span></span>

</div>

<h1>Orwell</h1>

<p>Данный сайт представляет из себя демоверсию системы управления робототехническим комплексом - Orwell</p>

</div>

</div>

<!-- /#page-content-wrapper -->

</div>

Код на JavaScript:

<!-- Nav Icon Script -->

<script>

$(document).ready(**function**(){

$('.nav-icon').click(**function**(){

$(**this**).toggleClass('open');

});

});

</script>

В итоге, шаблон сайта с открытой навигацией выглядит так:

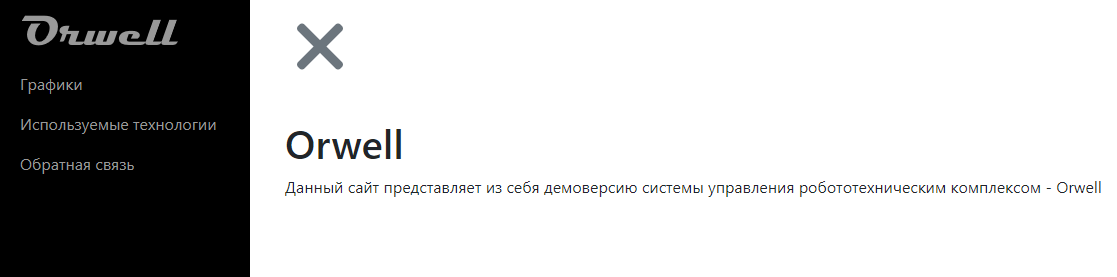
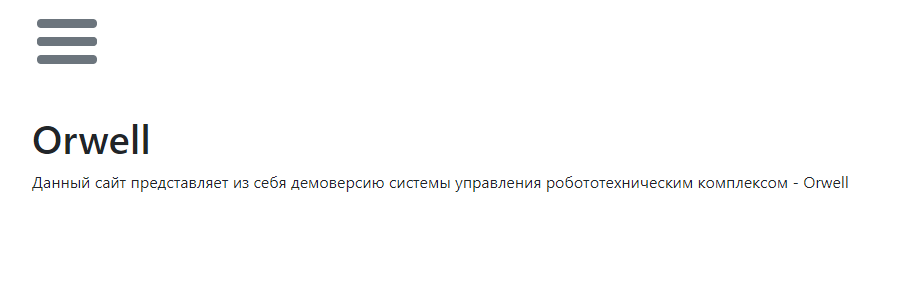


Рисунок 1 – Шаблон сайта с «открытой навигацией»

С закрытой навигацией:

Рисунок 2 – Шаблон сайта с «закрытой навигацией»

Также на данном сайте реализована функция обратной связи. Поскольку проект будет распространяться бесплатно и модифицироваться вместе с сообществом независимых разработчиков, очень важно иметь с ними обратную связь. В коде ниже можно посмотреть, как эта функция реализована через PHP:

<?php

/\*

\* CONFIGURE EVERYTHING HERE

\*/

// an email address that will be in the From field of the email.

$from = 'apgreeid@astropenguin.ru';

// an email address that will receive the email with the output of the form

$sendTo = 'apgreeidmikheev@gmail.com';

// subject of the email

$subject = 'New message from contact form';

// form field names and their translations.

// array variable name => Text to appear in the email

$fields = **array**('name' => 'Name', 'surname' => 'Surname', 'phone' => 'Phone', 'email' => 'Email', 'message' => 'Message');

// message that will be displayed when everything is OK :)

$okMessage = 'Письмо успешно отправлено! Спасибо за ваше обращение, в ближайшее время мы вам ответим с:';

// If something goes wrong, we will display this message.

$errorMessage = 'При отправке произошла ошибка. Пожалуйста, попробуйте позже.';

/\*

\* LET'S DO THE SENDING

\*/

error\_reporting(**0**);

**try**

{

**if**(count($\_POST) == **0**) **throw** **new** \Exception('Form is empty');

$emailText = "You have a new message from your contact form**\n**=============================**\n**";

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

// If the field exists in the $fields array, include it in the email

**if** (isset($fields[$key])) {

$emailText .= "$fields[$key]: $value**\n**";

}

}

// All the neccessary headers for the email.

$headers = **array**('Content-Type: text/plain; charset="UTF-8";',

'From: ' . $from,

'Reply-To: ' . $from,

'Return-Path: ' . $from,

);

// Send email

mail($sendTo, $subject, $emailText, implode("**\n**", $headers));

$responseArray = **array**('type' => 'success', 'message' => $okMessage);

}

**catch** (\Exception $e)

{

$responseArray = **array**('type' => 'danger', 'message' => $errorMessage);

}

// if requested by AJAX request return JSON response

**if** (!**empty**($\_SERVER['HTTP\_X\_REQUESTED\_WITH']) && strtolower($\_SERVER['HTTP\_X\_REQUESTED\_WITH']) == 'xmlhttprequest') {

$encoded = json\_encode($responseArray);

header('Content-Type: application/json');

**echo** $encoded;

}

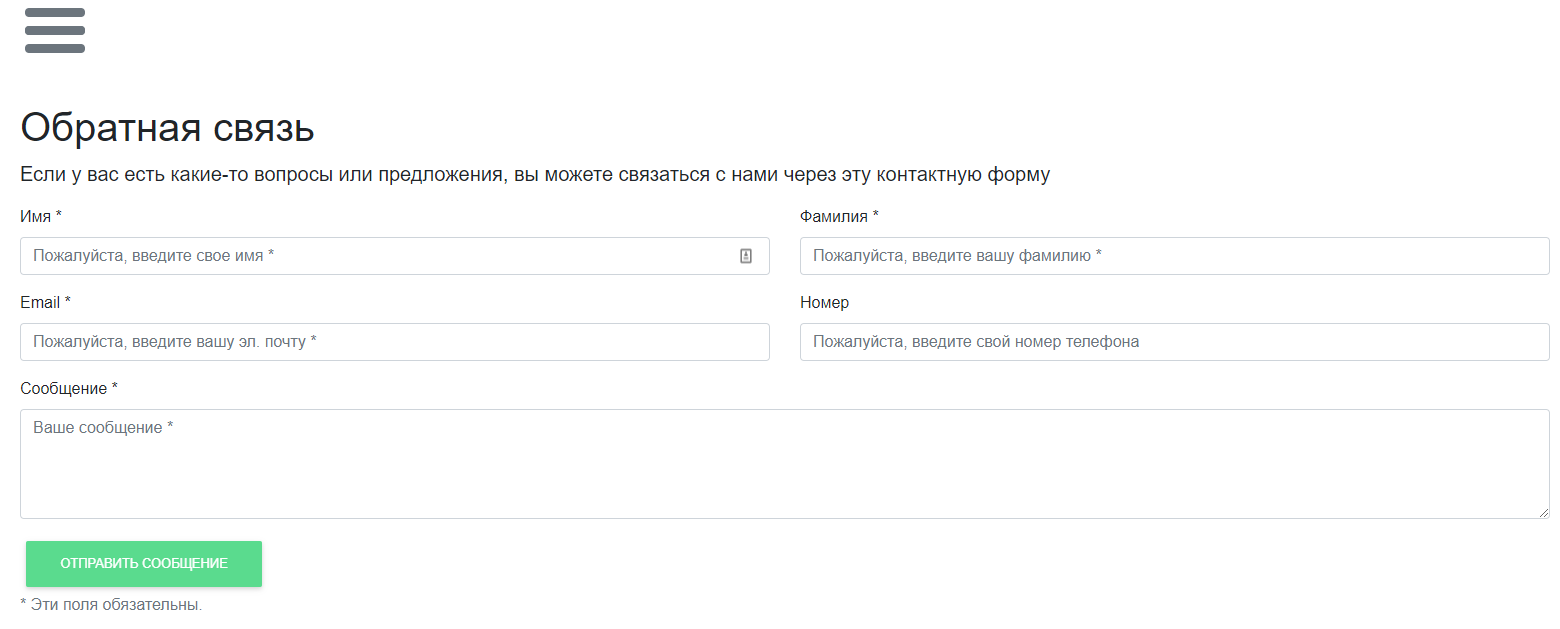
// else just display the message

**else** {

**echo** $responseArray['message'];

}

Страница с обратной связью выглядит так, как показано на рисунке ниже:

Рисунок 3 – Страница с формой обратной связи

# **Заключение.**

**Twitter Bootstrap -**  это замечательный фреймворк для создания современных, кросс-браузерных и стандартизованных интерфейсов. Продуманная структура кода HTML, JavaScript и CSS дает вам возможность создавать множество самых разнообразных элементов интерфейса и сетку сайта.

Свидетельством того, что Bootstrap это очень удачная разработка, является его непрерывно растущая популярность. Нынешний уровень развития этого фреймворка уже позволяет полностью создать любой web-интерфейс.

# **Список литературы**

1. Bootstrap: http://bootstrap-ru.com/index.php

2. Как сверстать веб страницу на Bootstrap: https://habrahabr.ru/post/211032/